Secure Audit Logging Systems

with Privacy Preserving

Richard Kramer, Member IEEE – Oregon State University

How does someone know they have been HACKED!?… and WHO did it!?

??????????

Audit Logs in the News!

"An audit trail that was maintained by the database company NGP VAN appears to show that four Sanders staffers conducted 25 specialized searches of the Clinton campaign's data, including queries for "turnout" and "primary priority" in a 40-minute window."

Audit Logs in the News!

Privacy & Security

EHR audit catches snooping employee

Nearly 900 notified of new HIPAA bre

By Erin McCann (/author/erin-mccann) | Jan

Health Information Technology

19 latest healthcare data breaches

Written by Akanksha Jayanthi (Twitter | Google+) | September 01, 2015 | Print | Email

Privacy & Security

Snooping employees disciplined after HIPA

'Appropriate actions have been taken w

By Erin McCann (/author/erin-mccann) | August 21, 20

Comp & Legal

Employee sacked after snooping patient EMR records

University Hospitals notifies patients of HIPAA breach

By Erin McCann (/author/erin-mccann) | December 02, 2014 | 10:51 AM

> "The incident was discovered after the hospital conducted an **EHR [Electronic Health Record] audit** back in October 2014. When it was first discovered only 14 individuals had had their PHI compromised."

Contributions / Agenda:

- **Provide a survey of Secure Audit Logging and review some important foundational work:**
  - **Schneier [3], Crosby [4], Goyal [5],**

- **Provide a detailed review of recent key publications:**
  - **Privacy preserving security -** Gunnar Hartung, "Secure Audit Logs with **Verifiable Excerpts** – Full Version", ACM, International Association for Cryptologic Research, 2016 [6,7]

  - **Multi-level user security with privacy preserving -** Se Eun Oh, et al., "**Privacy-preserving audit for broker-based health exchange"** ,ACM, Proceedings of the 4th ACM conference on data and application security and privacy, 2014 [8,9]

- **Identify potential Future Work and applications for the benefit of Audit Logging for EHR (Electronic Health Records) related events**

- **Provide an up-to-date list of Audit Logging tools and systems… some of them are FREE!** [10]

What is an Audit Log?

Secure Audit Logs

… are logs that **securely store security related information and events.**

Audit Logs are required by the government:
- Healthcare (HIPAA)
- Financial
- Legal
- Privacy Regulations

Examples include [1]:

- Reading critical files
- Account changes
- OS changes
- Major application changes
- Remote access
- Application transactions such as recording the sender / recipients of emails

What Generates an Audit Log?

- Audit Logs are generated from a wide variety **of aggregated sources** including antivirus software, firewalls, intrusion detection systems**, policy making systems** [8], and the like.

Example [2]:

Intrusion Detection System

```
[**] [1:1407:9] SNMP trap udp [**]
[Classification: Attempted Information Leak] [Priority: 2]
03/06-8:14:09.082119 192.168.1.167:1052 -> 172.30.128.27:162
UDP TTL:118 TOS:0x0 ID:29101 IpLen:20 DgmLen:87
```

Personal Firewall

```
3/6/2006 8:14:07 AM,"Rule ""Block Windows File Sharing"" blocked (192.168.1.54,
netbios-ssn(139)).","Rule ""Block Windows File Sharing"" blocked (192.168.1.54,
netbios-ssn(139)).  Inbound TCP connection.  Local address,service is
(KENT(172.30.128.27),netbios-ssn(139)).  Remote address,service is
(192.168.1.54,39922).  Process name is ""System""."

3/3/2006 9:04:04 AM,Firewall configuration updated: 398 rules.,Firewall configuration
updated: 398 rules.
```

Antivirus Software, Log 1

```
3/4/2006 9:33:50 AM,Definition File Download,KENT,userk,Definition downloader
3/4/2006 9:33:09 AM,AntiVirus Startup,KENT,userk,System
3/3/2006 3:56:46 PM,AntiVirus Shutdown,KENT,userk,System
```

Antivirus Software, Log 2

```
240203071234,16,3,7,KENT,userk,,,,,,,16777216,"Virus definitions are
current.",0,,0,,,,,0,,,,,,,,,,SAVPROD,{ xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx },End
User,(IP)-192.168.1.121,,GROUP,0:0:0:0:0:0,9.0.0.338,,,,,,,,,,,,,,
```

Antispyware Software

```
DSO Exploit: Data source object exploit (Registry change, nothing done)  HKEY_USERS\S-
1-5-19\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\0\1004!=W-3
```

It's not enough to simply have an Audit Log

The Audit Log needs to be secure.

Securing Audit Logs is of the utmost importance because "Bad guys" seek to cover up their malicious activity.

Ideally -

1) We can ***prevent alteration*** of the logs
2) We can ***verify, via analysis*** that the logs have not been changed
3) We **only decrypt portions of the log to preserve privacy**

The objective of Secure Audit Logging Systems is to protect Audit Logs from being compromised.

Overview of the Art

Historically, a number of foundational papers have considered various systems to ensure the privacy and security of Audit Logs:

- **Schneier (1999),** "***Secure Audit Logs to Support Computer Forensics***" – Provides methods and systems for **protecting** an Audit Log such that the Audit Log is secure, even if the server that the Audit Log resides on, is compromised [3].
- **Crosby et al (2009)** – "Efficient Data Structures for Tamper-Evident Logging". In short, Crosby introduced efficient data structures for ***tamper-evident logging*** [5] - only parts of the data is exposed [4], **thus protecting private information.**
- **Goyal et al (2006),** "***Attribute-based Encryption*** for fine-grained **access control** of encrypted data". Protects privacy of the information in the Audit Log **based on attributes and user access levels** [5].

Overview of the Art - ***Securing Audit Logs***

Schneier uses a "Hash Chain", where new entries added to the log are hashed on top of previously hashed log entries [3].

- Thus if a "bad guy" that took over a log server at some time, Y_j, he could not go back and alter the log at time Y_{j-1} and before

Time j-1

Time j

Overview of the Art - ***Securing Audit Logs***

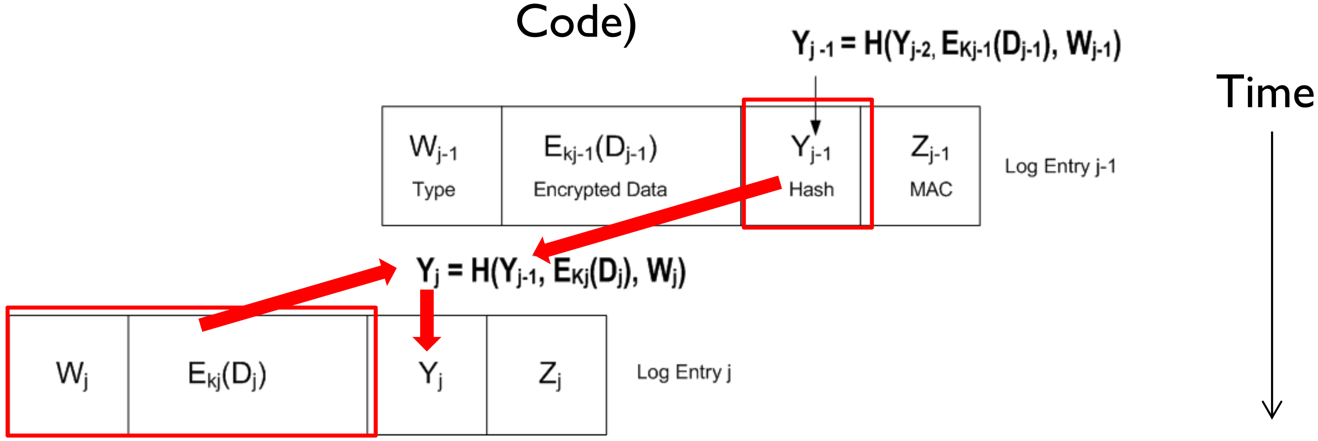
Schneier "Hash Chain":

$$Y_j = H(Y_{j-1}, E_{Kj}(D_j), W_j), \text{ where } Y_{j-1} \text{ is based on } Y_{j-1} = H(Y_{j-2}, E_{Kj-1}(D_{j-1}), W_{j-1})$$

and so on.

Where:

- *W* = log entry type (e.g., File Accessed, Permissions changed, etc.)
- *D* = log entry data
- *Y* = hash chain entry
- *Z* = MAC (Message Authentication Code)



Overview of the Art - ***Detecting Tampering of an Audit Log***

Crosby et al (2009) – "Efficient Audit Logs with Verifiable Excerpts" [4].

In short, Crosby introduced efficient data structures for tamper-evident logging [4].

- Crosby taught that it was pointless to have tamper resistant logs, if nobody ever looks at the logs to determine if they have been tampered with. Thus Crosby developed "***tamper evident logs***"

Thus:

- Crosby introduced the notion of a "**commitment**" which he calls a "**snap shots**" of the Audit Log up to a certain point in time

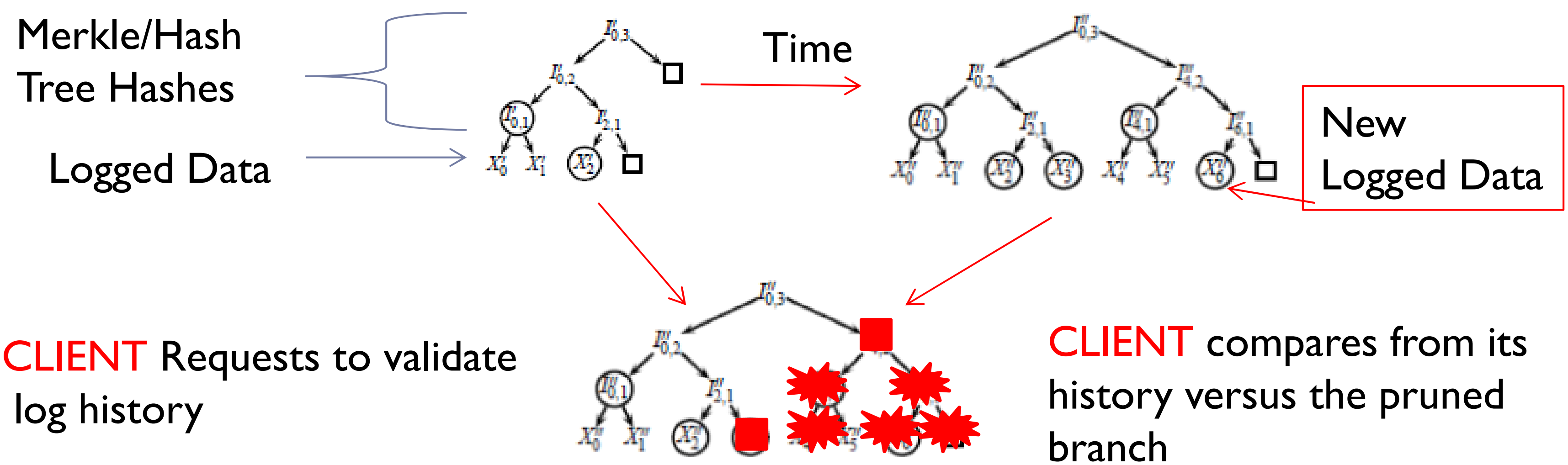
- Crosby assumes an "***untrusted logger***", *where he used the* ***clients*** to verify that the "commitments" being provided by the logger are true

Overview of the Art - ***Detecting Tampering of an Audit Log***

Crosby method in a nutshell:

- The "tamper evident log" is based on **Merkle trees**, where the leaves represented the data (events), and the roots contain hashes

Tree (or part of it) = a tamper evident summary of the data



Take new tree, delete nodes and rebuild – Do old (saved) and rebuilt hashes match?

[4]

Overview of the Art - ***Detecting Tampering of an Audit Log***

Crosby method in a nutshell:

- The Merkle Tree nodes are essentially a series of **one-time signatures** **(i.e., Lamport, etc.)**

- Only data from **"pruned trees"** that contain the portion of the tree structure and related hashes being checked needs to be sent/checked

Crosby further provides:

- ***Privacy preserving ("Private" search) by Audit Logging and exposing attributes about an event***, but not the entire event contents itself

Overview of the Art - ***Hierarchical Identity-Based Encryption for Audit Logs***

Goyal et al (2006), "Attribute-based Encryption for fine-grained access control of encrypted data" [5].

Goyal uses ***"Hierarchical Identity-Based Encryption (HIBE). HIBE*** *p*rovides the ability to selectively decrypt Audit Log "***attributes***" based on the access control level privileges granted to a specific user [5].

- ***Thus provides privacy at a hierarchical access control level.***

For example, the following ***attributes*** may have different access control levels, or overlapping access control levels, so that users may or may not decrypt some or all of the information:

- Name
- Date
- Source IP address
- Destination IP address
- Protocol
- Or other attribute based data

Goyal's implementation is based on a "tree structure" where Goyal ***called the attributes "***leaves", and the nodes of the tree consisted of logical "AND"s and "OR"s related to ***access right privileges*** (e.g., based on leaves, a user is logically allowed or denied access).

Overview of the Art - ***Hierarchical Identity-Based Encryption for Audit Logs***

Goyal's Encryption/Decryption key allows privacy for a specific set of attributes, thus ***preserves privacy by limiting access*** to Audit Log data by those not authorized to see specific attributes:

$$D = f(M, Pk, \gamma_1 \ldots \gamma_n)$$

Where:

D = Decryption Key,

M= Message

Pk is the public key information generated from a Master Key (*MK*)

$\gamma_1 \ldots \gamma_n$ are the attributes (file accessed, OS system configuration changed, whatever….)

Pro:

- Provides some elements of ability to search on encrypted data (attributes) and privacy for the encrypted Message *M* and access level.

Con:

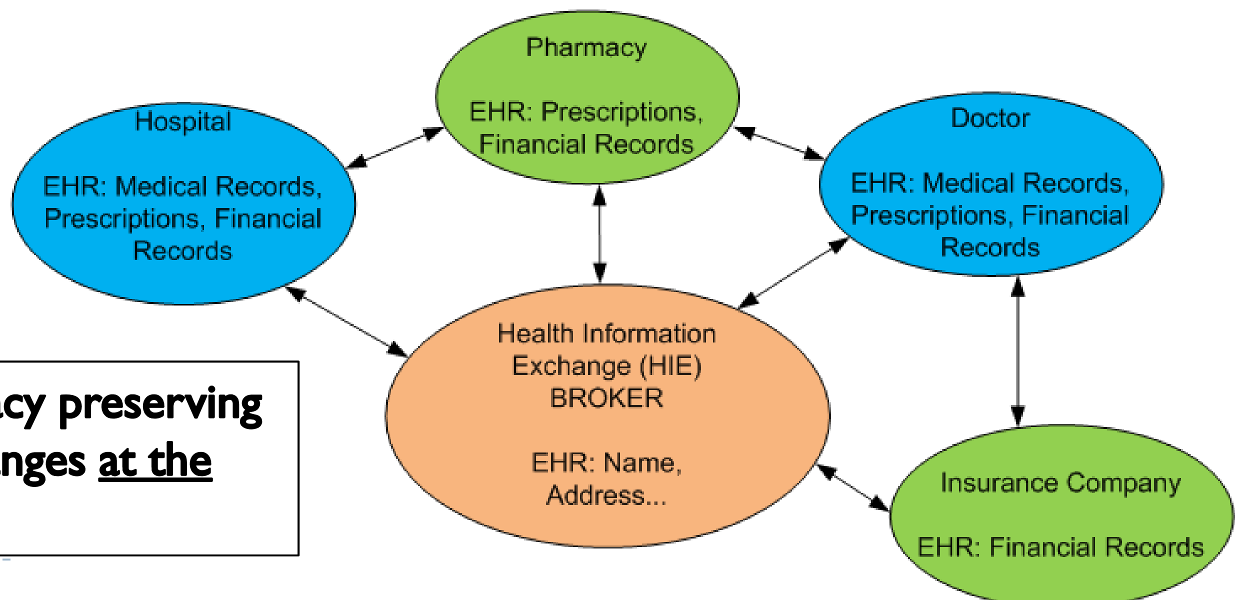
- While at the same time the disadvantage of the system is that ***the set of attributes is sent in clear text.***

The Current State – Secure Audit Logging Systems with Privacy Preserving

Hartung (2014) – ***Builds on Crosby:*** “Secure Audit Logs with Verifiable Excerpts” or “SALVE” for short [6].

> **A good paper that addresses privacy preserving <u>at the security level to augment the “Oh et al” paper (shown below)</u>**

Oh et al (2014) – “Privacy Preserving audit for broker based health information exchanges” [8].



> **A good paper on privacy preserving for Health Care Exchanges <u>at the application level</u>**

The Current State of the Art for Secure Audit Logging Systems – ***Privacy Preserving***

Hartung (2014) – ***Builds on Crosby:*** "Secure Audit Logs with Verifiable Excerpts" or "SALVE" [6].

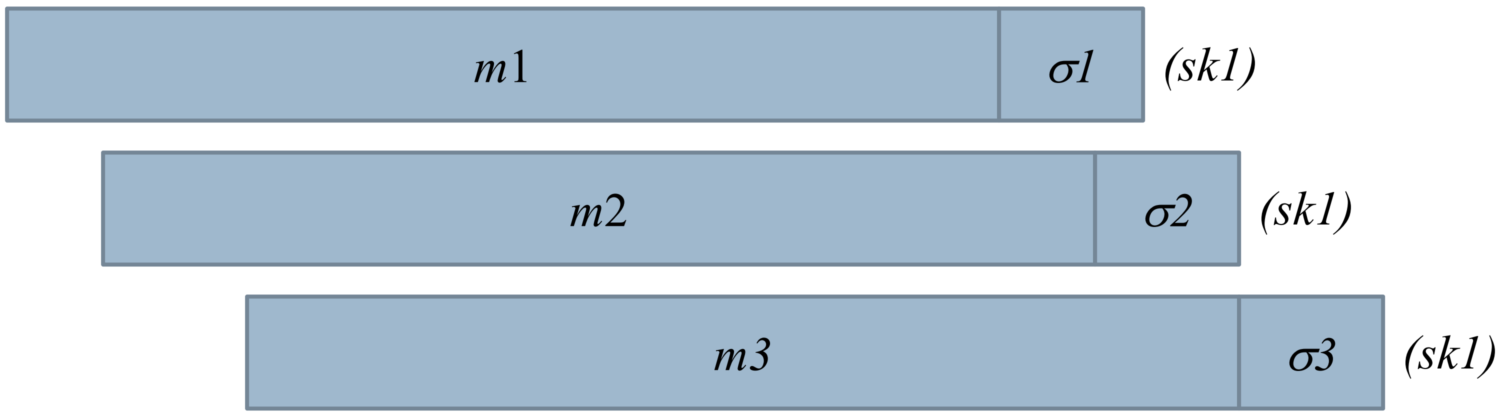
Hartung provides:

1) **Verification** - of an "Excerpt" is provided for BOTH:

- Completeness
- Correctness

2) Privacy preserving - in that only "Excerpts" of the log are audited, ***thus the remainder of the Audit Log remains private.***

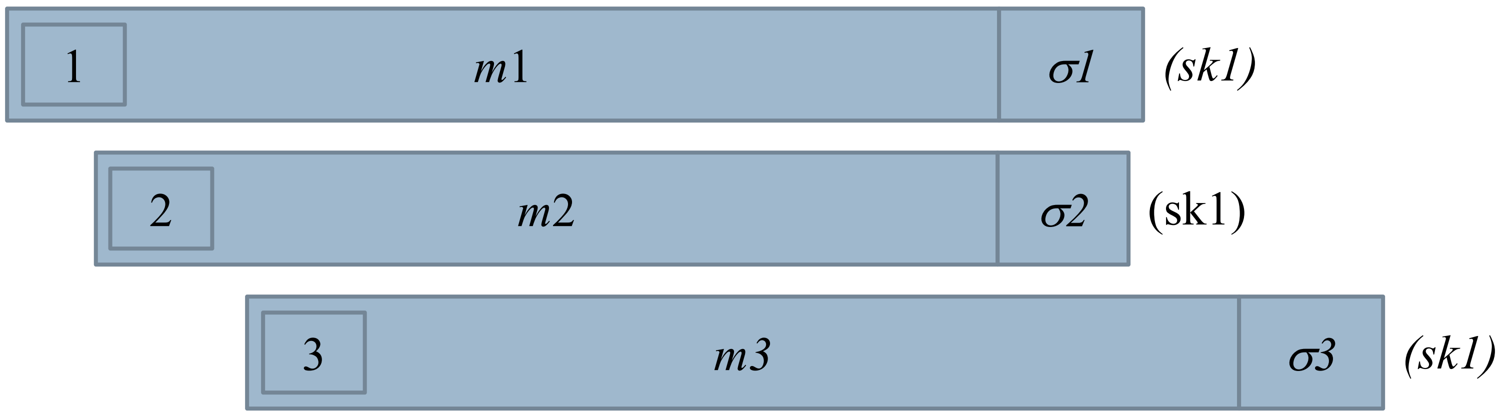
Audit Logging System Compromises



Schemes to secure Audit Logs using signatures have been broken

… and schemes using secret keys (sk) have been broken [6 at pg. 6]

Audit Logging System Compromises



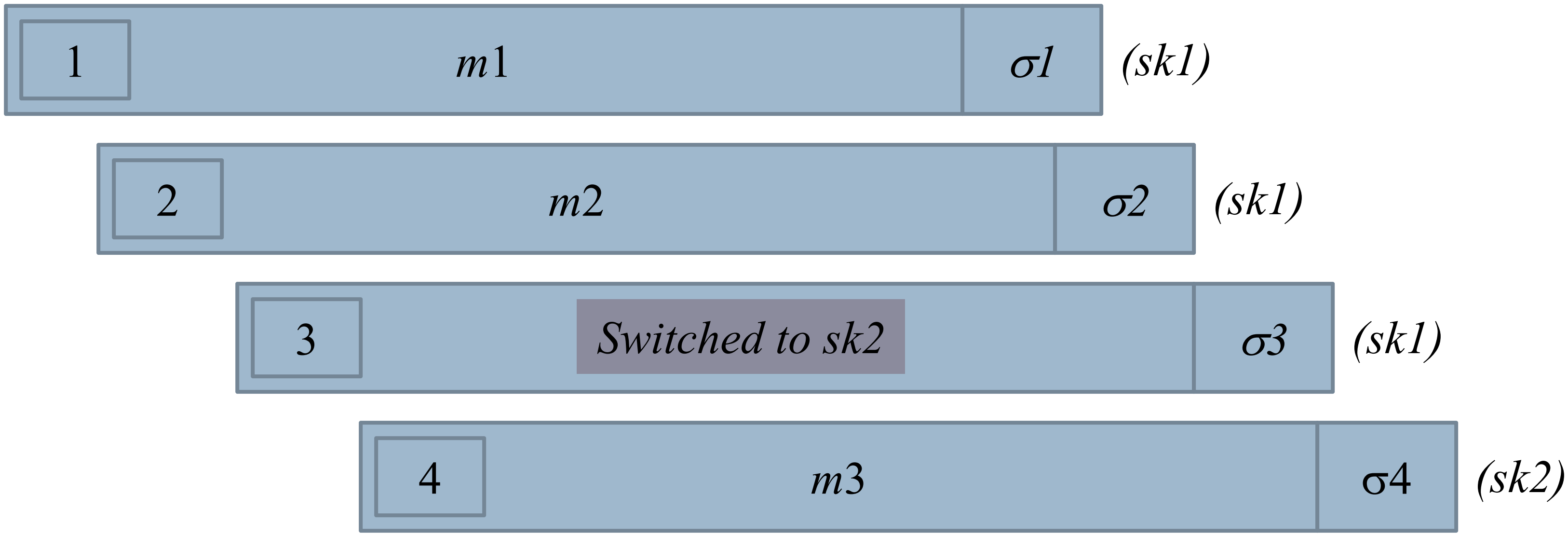
Schemes to secure Audit Logs using signatures have been broken

… and schemes using secret keys (sk) have been broken [6 at pg. 6]

And the removal of log entries / tricks to accept modified logs or reordering message attacks are known [*id.*]

So counters and epoch markers have been added [*id.*]

Audit Logging System Compromises



Schemes to secure Audit Logs using signatures have been broken

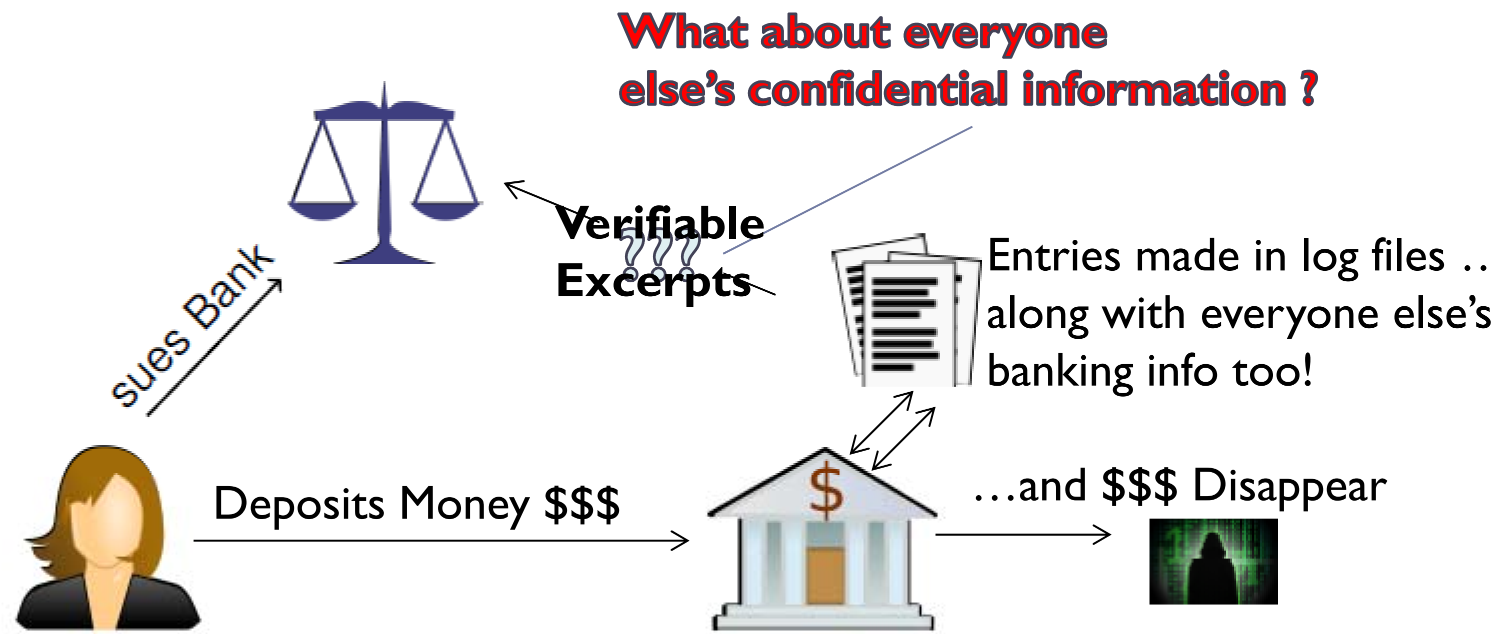
… and schemes using secret keys (sk) have been broken [6 at pg. 6]

And the removal of log entries / tricks to accept modified logs or reordering ordering attacks are known [*id.*]

So counters and epoch markers have been added [*id.*]

And yet still, truncation attacks exist [*id.*]

The Current State of the Art for Secure Audit Logging Systems - ***Privacy Preserving***



- Hartung's verifiable "**<u>Excerpts</u>**" solves the problem. Excerpts are Audit Log records that entail specific:
  - **"Categories" (e.g., Bank Account Opened, Deposit Make, Name, etc.)**
  - **Epochs (T states) from one Audit Log message(s) entry state to the next**

[7]

Security Scheme

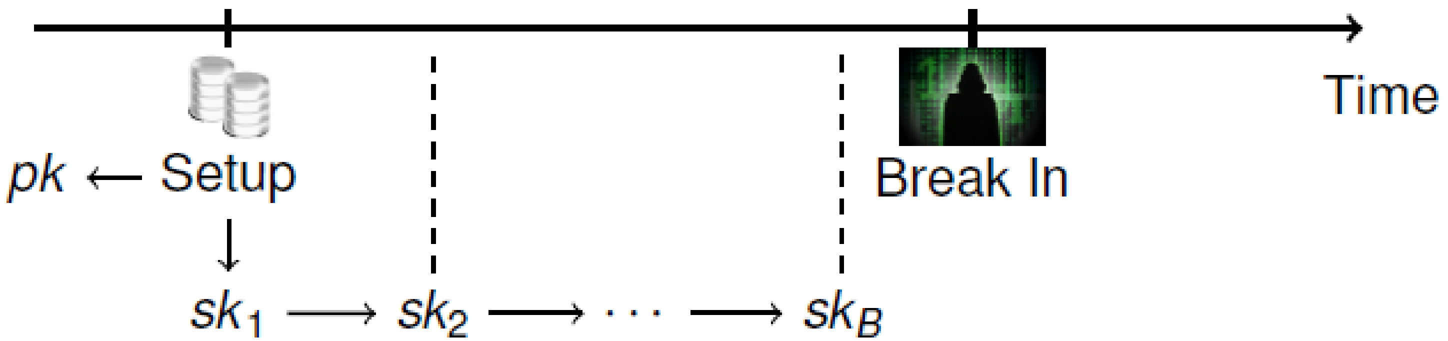
In contrast to Hash Chains, **<u>Hartung "chains" together Secret Keys that are then used to create unique signatures</u>**

- Each new "Secret Key" Sk_i at state "i" is based on the prior Secret Keys (Sk_{i-1}, Sk_{i-2} ...) where previous keys are DELETED:

$$\boldsymbol{Sk_i = f\,(\cancel{Sk_{i-1}}, \cancel{Sk_{i-2}}, \cancel{Sk_{i-3}}, \cancel{Sk_{i-4}} \ldots)}$$

Security Scheme

Hartung calls his cryptology scheme a "Categorized Key-Evolving Audit Log Scheme"



[7]

Security Scheme

When verification of an excerpt is desired, the functions "EXTRACT" and "VERIFY" respectively create a unique signature for an excerpt and verifies the integrity of the excerpt as follows:

$\boldsymbol{\sigma}'$, ***Excerpt*** $\leftarrow$ **EXTRACT(*sk_i*, *$M_{o,j}$*, *$\sigma_{0,j}$*, *V*),** where "Extract" also produces a unique ***pk*** **(Public Key).**

True / False $\leftarrow$ VERIFY(*pk*, *V*, *Excerpt*, *σ'*)

Where:

- sk_i is the Secret Key for epoch i
- $M_{o,j}$ is the Message Log excerpt
- $\sigma_{0,j}$ is the previous signature (σ_0 ... σ_j) that is created for a specific excerpt
- σ_E is the excerpt signature generated using the private key sk_i
- $\sigma' = \sigma_{0...j},\ \sigma_E$
- V is a set of categories, named (v_0, … v_j) for the excerpt (Bank Account Opened, Deposit Make, Name, etc.)

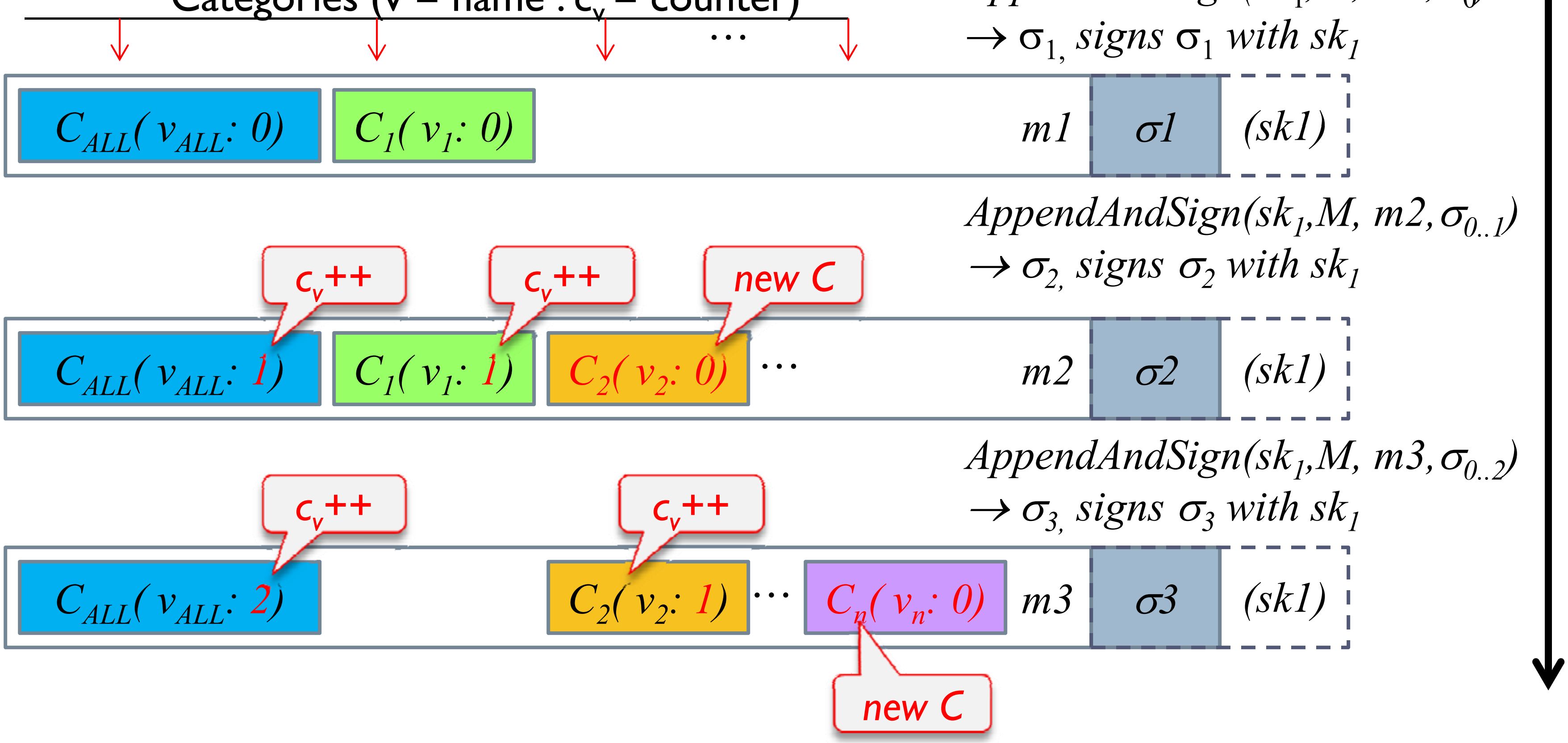
Example Scenario for Audit Log "*M*"

Time

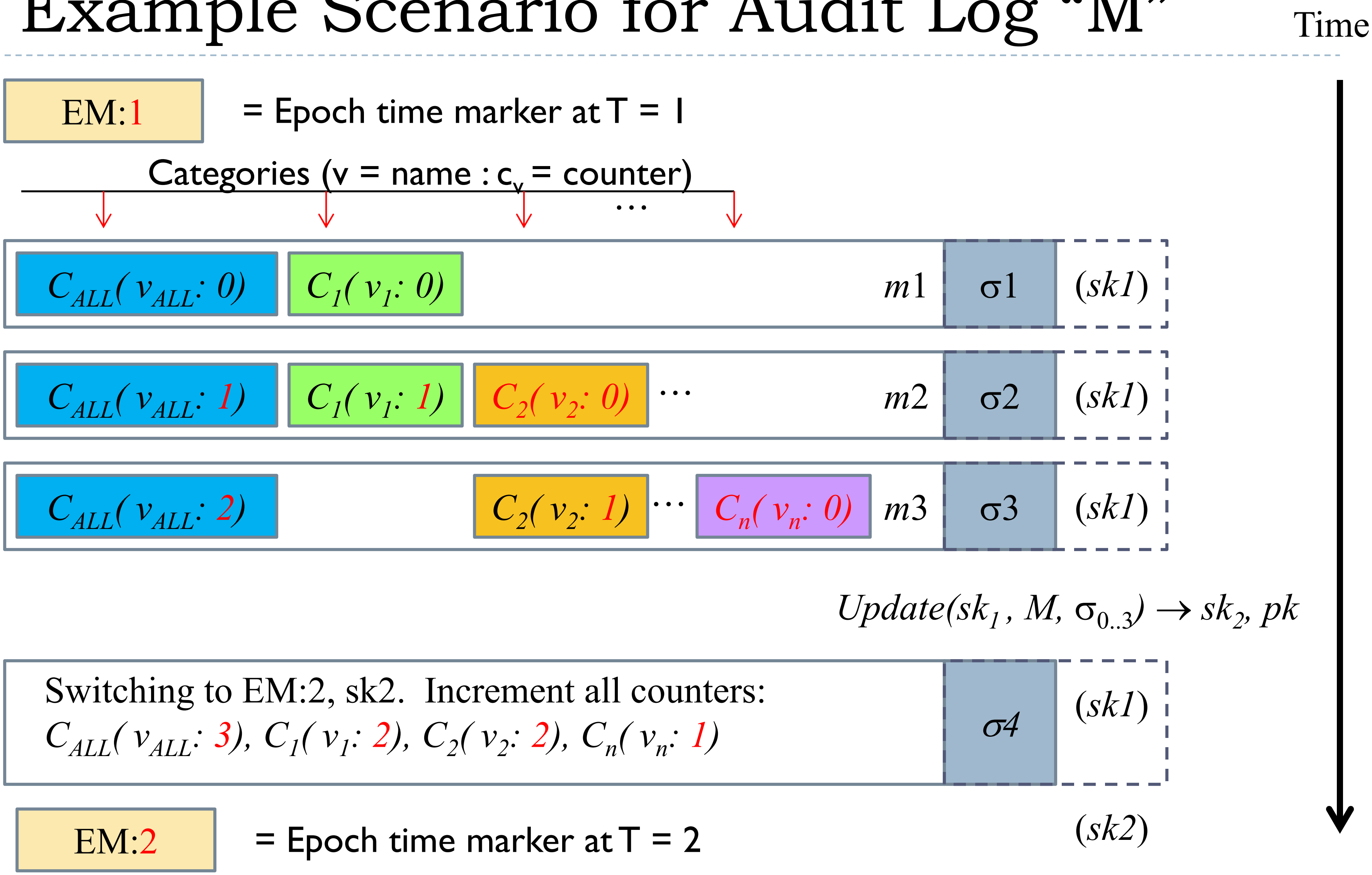
EM:0 = Epoch time marker at T = 0 — $KeyGen(T) \rightarrow sk_0, pk$

EM:1 = Epoch time marker at T = 1 — $Update(sk_0, M, \sigma_0) \rightarrow sk_1, pk$

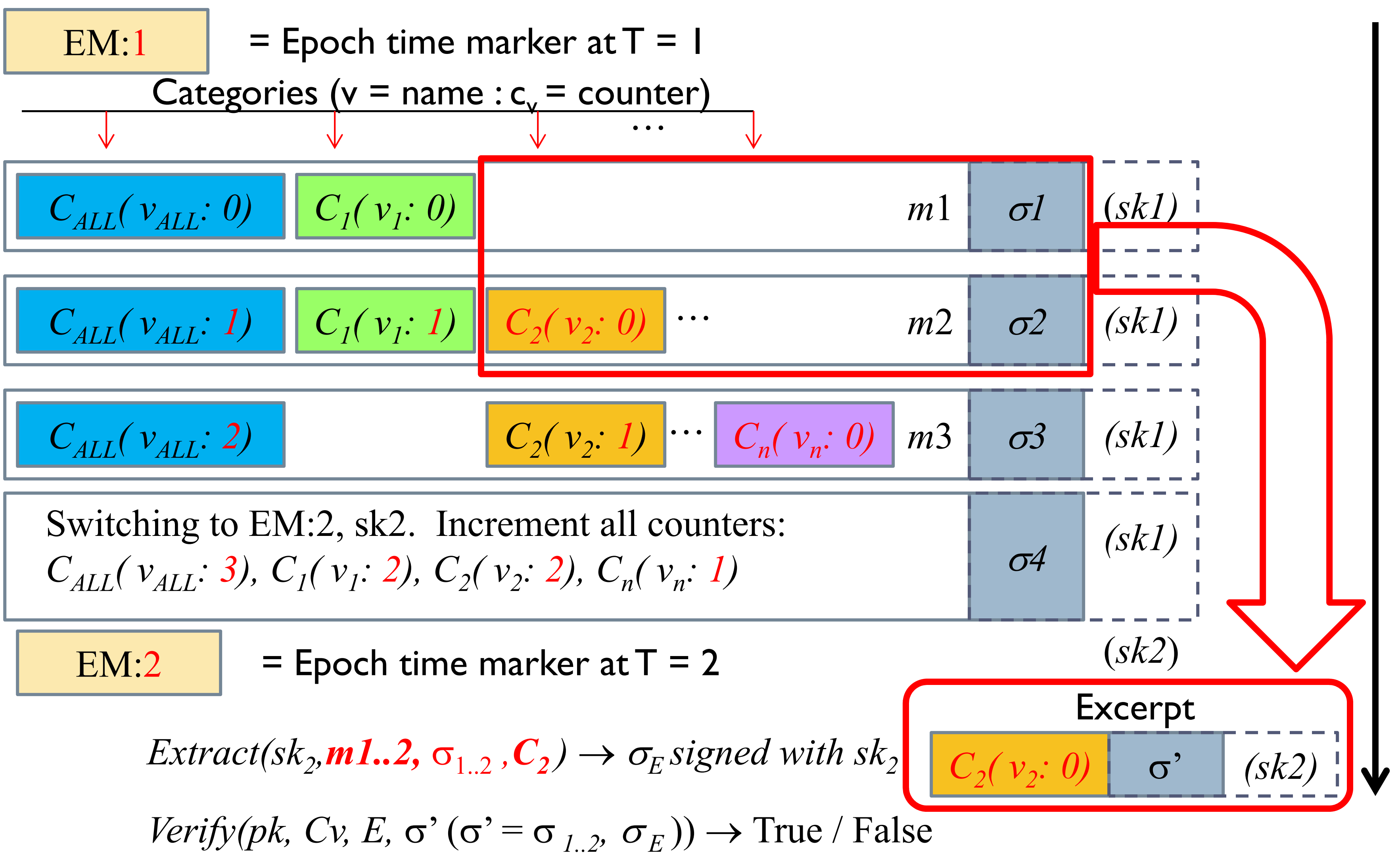
Categories (v = name : c_v = counter)



Example Scenario for Audit Log "M"



Example Scenario for Audit Log “M”



Performance

	Logcrypt	SALVE	Ma and Tsudik	BAF	LogFAS
Algorithm	**Runtime**				
Key Generation	KeyGen	KeyGen	KeyGen	$2T \times \text{ModExp} + 5T \times H$	$\text{KeyGen} + (T+1) \times \text{ModExp} + T \times \text{Sign}$
Log Entry Signing	Sign	Sign	Asig	$2 \times H + 2 \times \text{ModAdd}$	$1 \times H + 1 \times \text{ModExp} + 2 \times (\text{ModMul} + \text{ModAdd})$
Updating	$\text{KeyGen} + 1/n \times \text{Sign}$[11]	$\text{Update} + \text{Sign}$	Update	$2 \times H$	deletion only
Excerpt Signing	—	Sign	—	—	—
Verification	$\lvert M \rvert \times \text{Verify}$	$(\lvert E \rvert + i + 1) \times \text{Verify}$	Aver	$(\lvert M \rvert + 1) \times \text{ModExp} + (2\lvert M \rvert - 1) \times \text{ModMul}$	$2 \times \text{ModExp} + (\lvert M \rvert + 1) \times \text{ModMul}$
Datum	**Size**				
Secret Key	$\mathcal{O}(n) \times \lvert sk \rvert$	$\lvert sk \rvert$	$\lvert sk \rvert$	$4 \times \text{BigInt}$	$(T - i) \times (5 \times \text{BigInt} + \lvert \sigma \rvert)$
Public Key	$\lvert pk \rvert$	$\lvert pk \rvert$	$\lvert pk \rvert$	$(4T + 3) \times \text{BigInt}$	$4 \times \text{BigInt} + \lvert pk \rvert$
Log File Signature	$(\lvert M \rvert + i) \times \lvert \sigma \rvert + i \times \lvert pk$	$(\lvert M \rvert + i) \times \lvert \sigma \rvert$	$\lvert \sigma \rvert$	$2 \times \text{BigInt}$	$\lvert M \rvert \times (5 \times \text{BigInt} + \lvert \sigma \rvert)$
Excerpt Signature	—	$(\lvert E \rvert + i + 1) \times \lvert \sigma \rvert$	—		—

Pros:

- **Forward Integrity**
- **Privacy Preserving** when contrasting the entire Audit Log to an **Excerpt**

Cons:

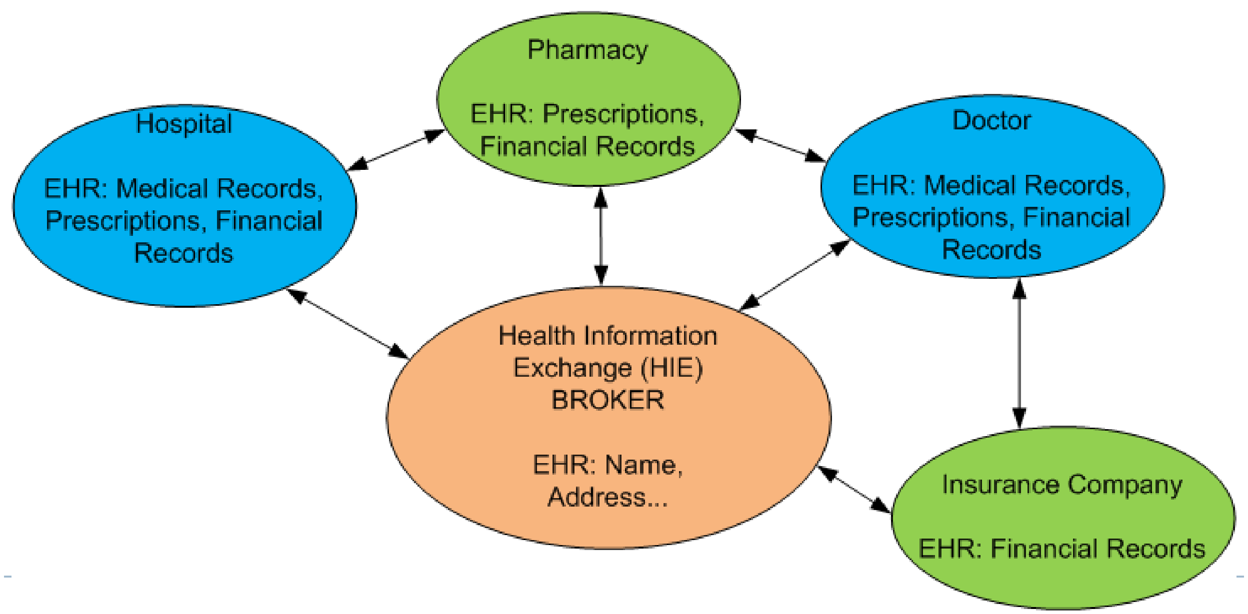
- Seemingly **large** signature Audit Log file signature and Excerpt signature which concatenate previous signatures and is a function of message size and categories
- **Slower computational** time as compared to the more efficient BAF, LogFAS approaches (ECE 599 – Winter 2017 term).

[6]

<u>**The Current State**</u> of the Art for Secure Audit Logging Systems – ***Health Care Exchanges***

Oh et al. provides a unique application using ***HIBE*** <u>***(Hierarchical Identity Based Encryption)***</u> – see Golay (2006) above

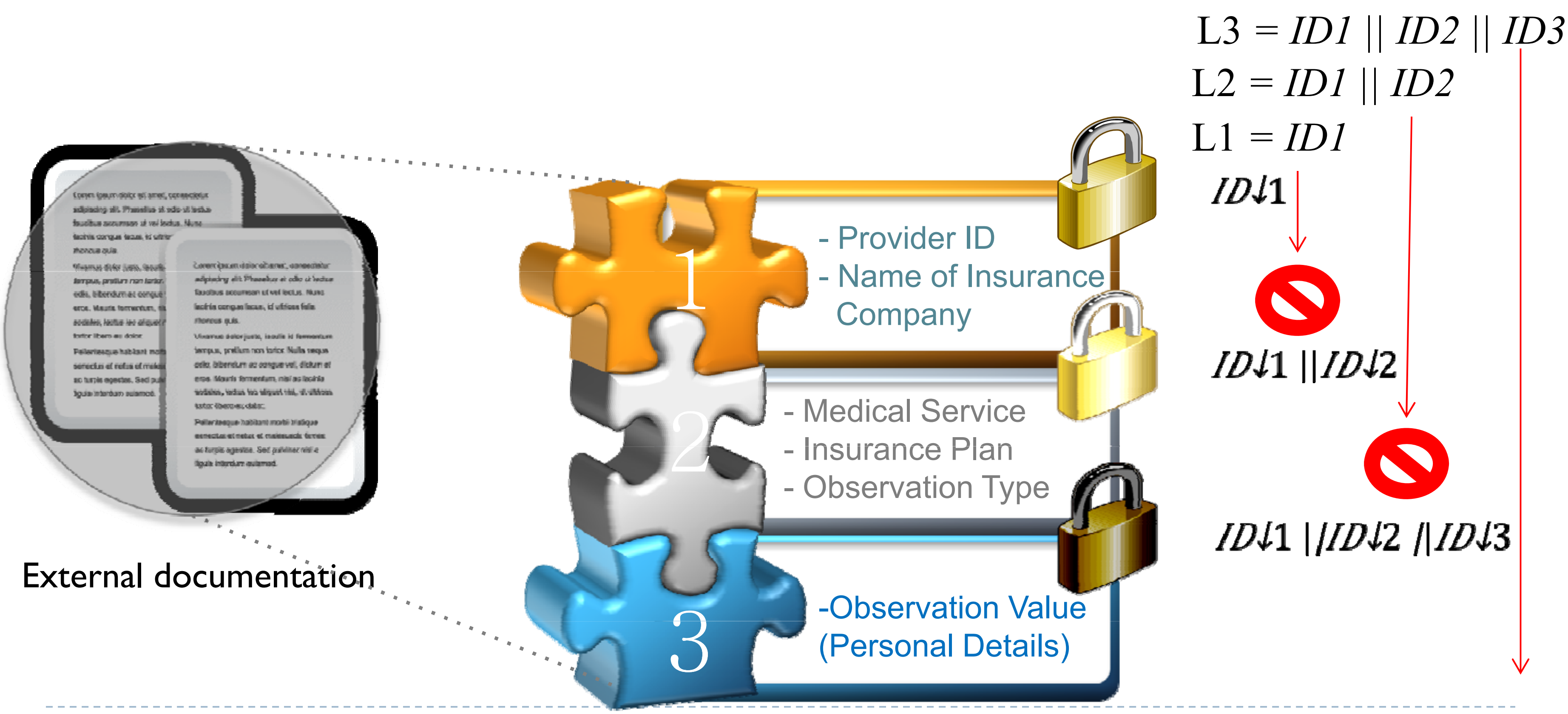
For the management and auditing of ***EHRs (Electronic Health Records)*** based on authorization "levels".



Privacy Preserving Data Management

- Enhance security with Hierarchical Identity based Encryption (HIBE) to allow limited access to relevant external documentation

Authorization Level



[8,9]

The Current State of the Art for Secure Audit Logging Systems *– **Health Care Exchanges***

- Access rights are accomplished in layers and embedded ***within the cryptography system:***

$$\textbf{\textit{Cipher Text Enc}}_{ID_i}(D_i) = \textbf{\textit{HIBE.Encrypt}}(Pub, ID_i, D_i)$$

Identity Level

Where:

- D_i = Data for a specific level (D_1 is least sensitive, D_n is most sensitive)
- ID_i = The identity level, where $ID_2 = id_1, id_2$, and so on
- pk = Public parameters generated at the same time the Master Key is generated during setup

Hierarchical Encryption

- Billing Table

		Security Level	Lowest		Highest
			Level 1 Provider ID and Insurance Company	Level 2 Medical Service Type, Insurance Plan and Observation Type	Level 3 Observation Value
patient-id	HCO-id	date-of-bill	level1	level2	level3
eeb728473e1949a	Carle07RQ12	2013:09:08:10:18:41	$Enc_{ID_{1,1}}(cs1010)$	$Enc_{ID_{1,2}}$(HEMOGLO...)	$Enc_{ID_{1,3}}(73.8)$
d99486a44ca64cb	Provena01AV98	2013:09:17:02:48:29	$Enc_{ID_{2,1}}(ra1010)$	$Enc_{ID_{2,2}}$(MCH,Auto...)	$Enc_{ID_{2,3}}(279)$
42210b2417d74b1	NWM0329W2	2013:10:21:11:47:22	$Enc_{ID_{3,1}}(pq1010)$	$Enc_{ID_{3,2}}$(PLATELET...)	$Enc_{ID_{3,3}}(11.6)$

Authorization Access Levels

$ID_{1,1}$

$ID_{1,2}$

$ID_{1,3}$

$ID_{row\#,\ level\#}$ = Patient ID || HCO ID || Date of Medical Bill || Sensitivity Level

$ID_{1,3}$ = eeb728473e1949a||Carle07RQ12||2013:09:08:10:18:41||level1

$ID_{1,2}$ = eeb728473e1949a||Carle07RQ12||2013:09:08:10:18:41||level1||level2

$ID_{1,1}$ = eeb728473e1949a||Carle07RQ12||2013:09:08:10:18:41||level1||level2||level3

The actual security methods to accomplish this are not addressed, thus Hartung and Goyal

[8,9]

The Current State of the Art for Secure Audit Logging Systems – ***Health Care Exchanges***

Implements "***Audit Trail and Node Authentication"*** (ATNA) as part of HIBE (Hierarchical Identify Based Encryption):

- ATNA not only logs events (e.g. a record has been accessed)
  - **Part of Audit system is to determine and log <u>WHY the record was accessed</u>**

- Uses an algorithm called **<u>REDUCE</u>**[1] to look for log violations based on a "policy formula"

1. D. Garg, L. Jia, and A. Datta, "Policy auditing over incomplete logs: theory, implementation and applications", ACM Proc. of CCS, 2011.

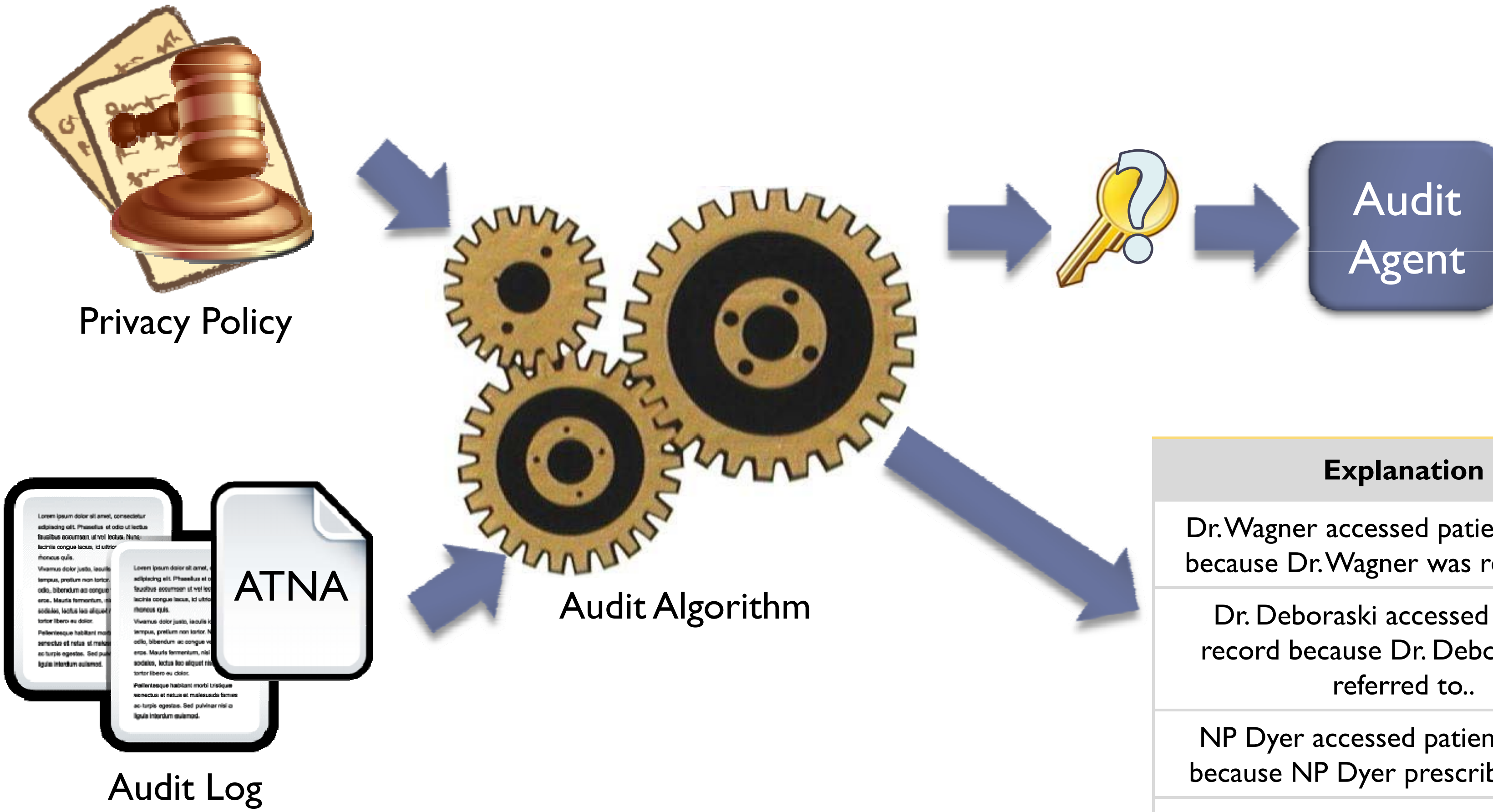
Uses **REDUCE** (basically an algorithm/policy language) with Explanations

$$
\begin{array}{llll}
\textit{Conj clause} & C & ::= & \bigwedge_i \varphi_i \\
\textit{Disj clause} & D & ::= & \bigvee_i \varphi_i \\
\textit{Formula} & \alpha & ::= & \langle \ell \rangle P \mid \langle \ell \rangle \top \mid \langle \ell \rangle \bot \mid \langle \ell \rangle C \mid \langle \ell \rangle D \\
 & & \mid & \langle \ell \rangle \forall \vec{x}.(c \supset \varphi) \mid \langle \ell \rangle \exists \vec{x}.(c \wedge \varphi) \\
 & & \mid & \sigma \rhd \varphi \\
\textit{Generalized form.} & \varphi & ::= & \alpha \mid \mathtt{expl}(\top, \gamma) \mid \mathtt{expl}(\bot, \gamma) \\
\textit{Explanation} & \gamma & ::= & \ell \mid \ell \circ \gamma \mid \gamma_1 \oplus \gamma_2 \mid \sigma \rhd \gamma
\end{array}
$$

[8,9]

Privacy Preserving Data Flow

- Verify legitimacy of access with logic-based audit algorithm



Explanation
Dr. Wagner accessed patient's record because Dr. Wagner was referred to..
Dr. Deboraski accessed patient's record because Dr. Deboraski was referred to..
NP Dyer accessed patient's record because NP Dyer prescribed patient
NP Dyer accessed patient's record because NP Dyer prescribed patient

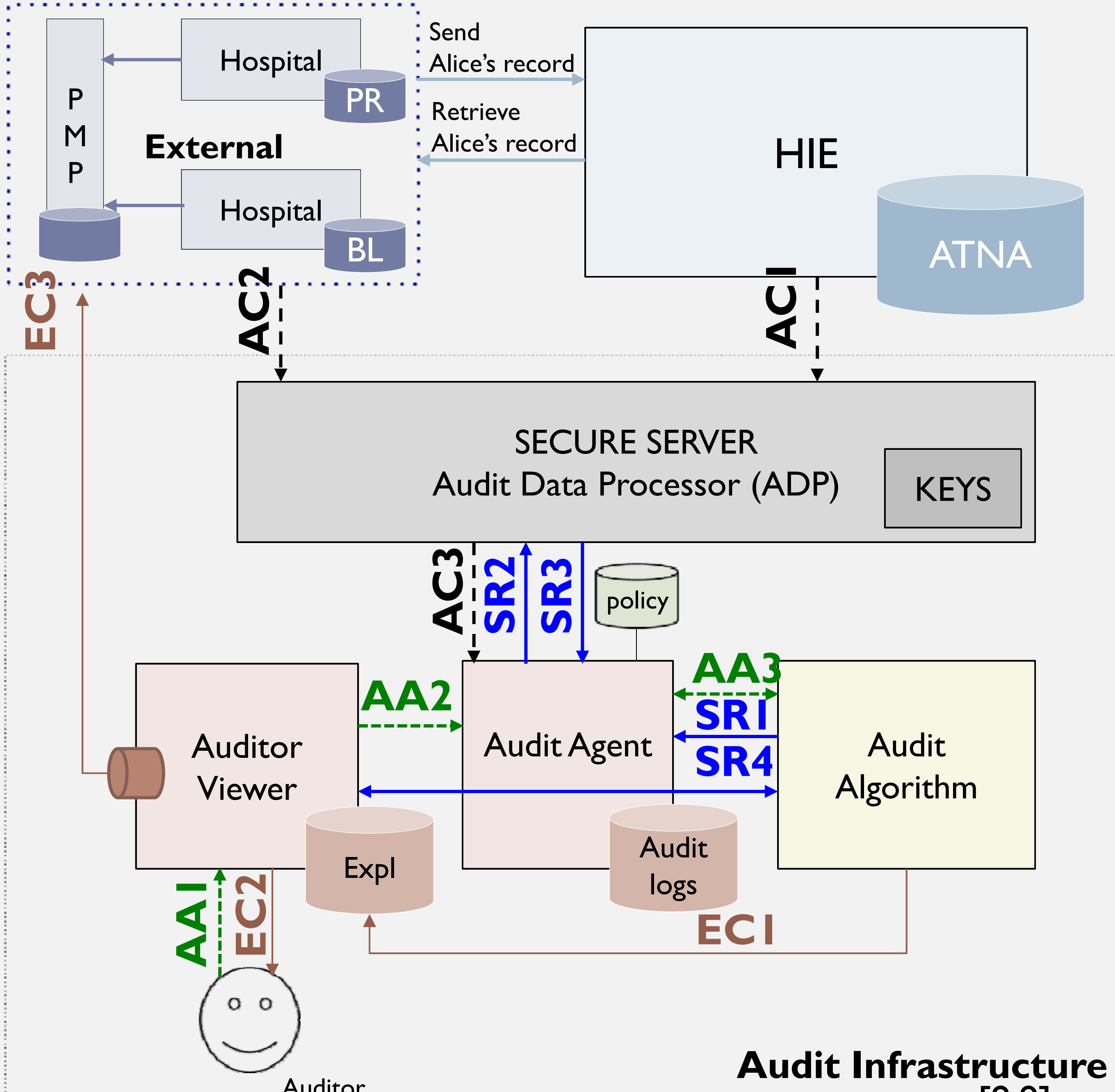
[8,9]

Audit data Collector (AC) Path: - - - - - - ->
1. ATNA logs
2. External documentations
3. Encrypted external doc and ATNA logs

Access Analysis (AA) Path: - - - - - - - - ->
1. Provider ID, Patient ID and Event time
2. Provider ID, Patient ID and Event time
3. SQLITE database

Supplement Resolution (SR) Path: ———>
1. Residual policy
2. ID(s)
3. Secret key(s)
4. SQLITE database

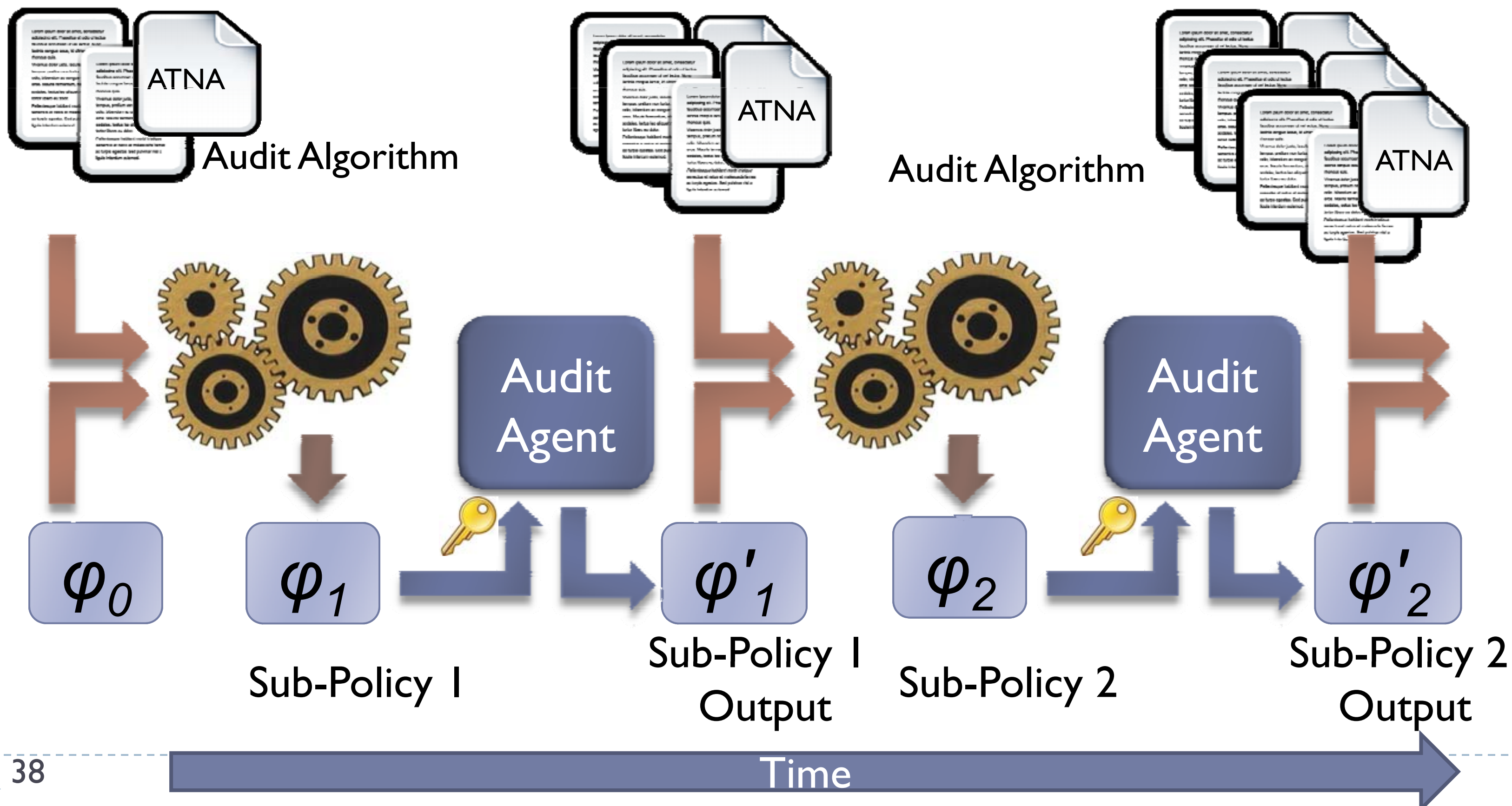
Explanation Creator (EC) Path: ———>
1. Explanations
2. Human-readable explanations
3. Report



Audit Infrastructure [8,9]

Audit Algorithm

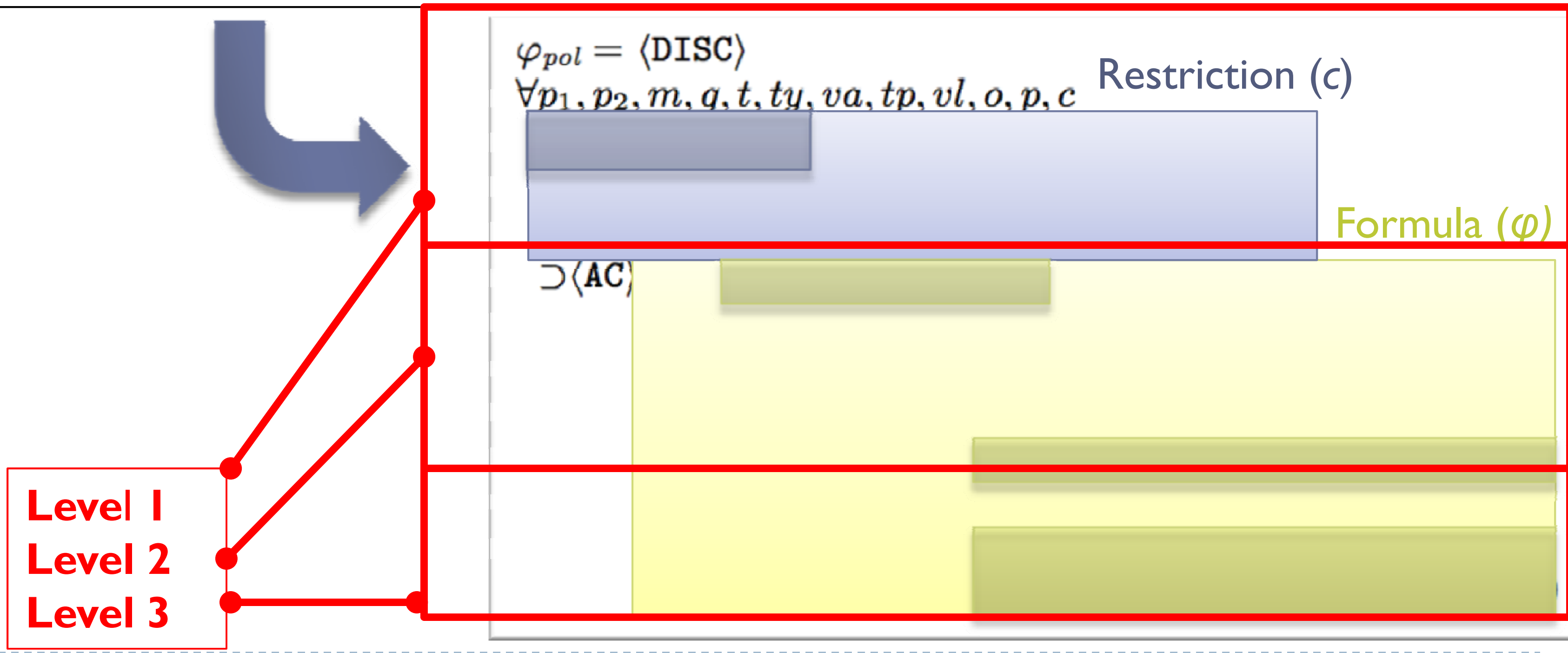
REDUCE[$\mathcal{L}$ (log), φ_n (privacy preserving policy n)] = φ'_n *(output)*



[8,9]

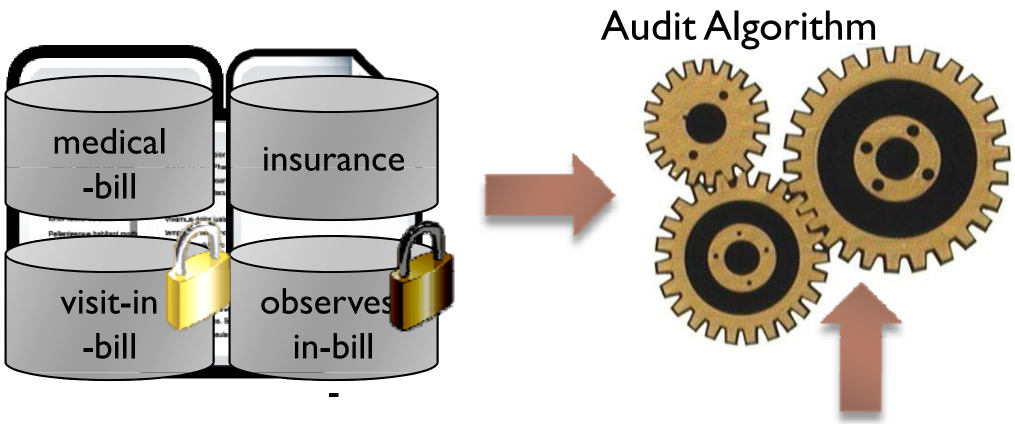
Policy Logic

Providers requesting or accessing to an EHR for "treatment" needs to be verified a relationship exists between p1, p2 and q and authorization level to see information in the medical bill.



[8,9]

Example Scenario



φ_0 =

∀p1,p2,m,q,t,ty,va,tp,vl,o,p,c
(send(p1, p2, m, t) ∧ tagged(m, q) ∧ includes(m,ty,va,t)
∧ patientInfo(q,tp,vl,t) ∧ organization(p2,o,t)
∧insuranceInfo(q,p,c,t))

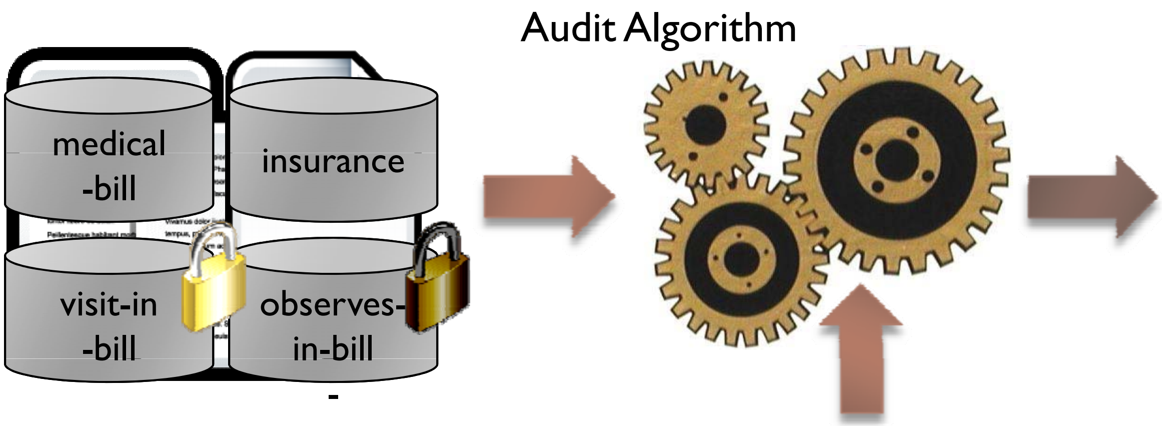
⊃ ∃t'. medical-bill(q,b,t')
∧ ((time-in(t,t',t+365)
∧ insurance(q,p,c,o,t')
∧ (visits-in-bill(q,p2,vl,o,t')
∨ observes-in-bill(q,p2,ty,va,o,t')))

{ p1→ Smith,
p2→ Kosta,
m → D1,
q → Grace,
t → DOA,
ty → NERVOUS SYS,
va → DEPRESSION,
tp → Service Type,
vl → O,
o → Carle,
p → PPO,
c → WA02,
t' → DOB }

Level 1
Level 2
Level 3

[8,9]

Example Scenario



φ_1 =
insurance(Grace PPO,
WA02,CA[ ]A)
∧(**visits-in-bill**(Grace,Kosta,O,
CARLE,DOB)
∨ **observes-in-bill**(Grace,Kosta,
NERVOUS SYSTEM,
DEPRESSION,CARLE,DOB))

level2

level3

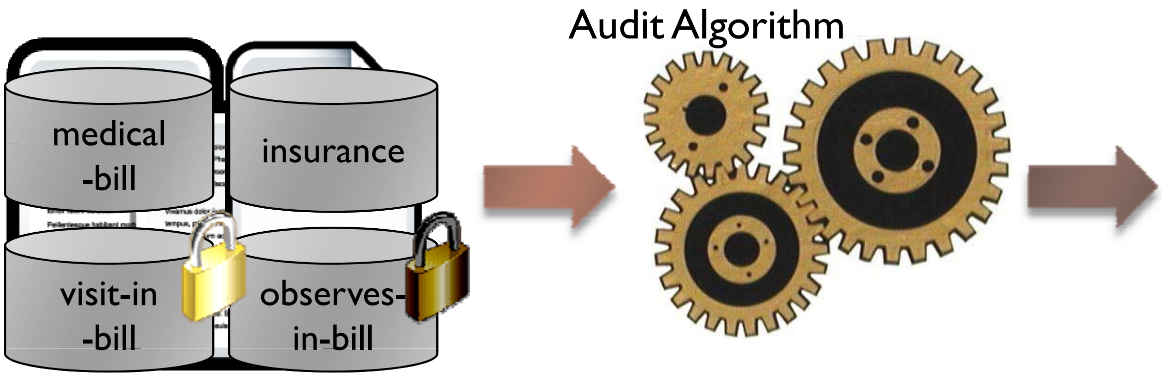
Audit
Agent

φ_0 =
∀p1,p2,m,q,t,ty,va,tp,vl,o,p,c
(send(p1, p2, m, t) ∧ tagged(m, q) ∧ includes(m,ty,va,t)
∧ patientInfo(q,tp,vl,t) ∧ organization(p2,o,t)
∧insuranceInfo(q,p,c,t))
⊃ ∃t'. medical-bill(q,b,t')
∧ ((time-in(t,t',t+365) ∧

φ_1 = ∃ insurance(q,p,c,o,t')
∧ (visits-in-bill(q,p2,vl,o,t')
∨ observes-in-bill(q,p2,ty,va,o,t')))

[8,9]

Example Scenario



$\varphi_1 =$
insurance(Grace,PPO,
WA02,CARLE,DOA)
∧(visits-in-bill(Grace,Kosta,O,
CARLE,DOB)
∨ observes-in-bill(Grace,Kosta,
NERVOUS SYSTEM,
DEPRESSION,CARLE,DOB))

$\varphi'_0 =$
∀p1,p2,m,q,t,ty,va,tp,vl,o,p,c
(send(p1, p2, m, t) ∧ tagged(m, q) ∧ includes(m,ty,va,t)
∧ patientInfo(q,tp,vl,t) ∧ organization(p2,o,t)
∧insuranceInfo(q,p,c,t))
⊃ ∃t'. medical-bill(q,b,t')
∧ ((time-in(t,t',t+365)

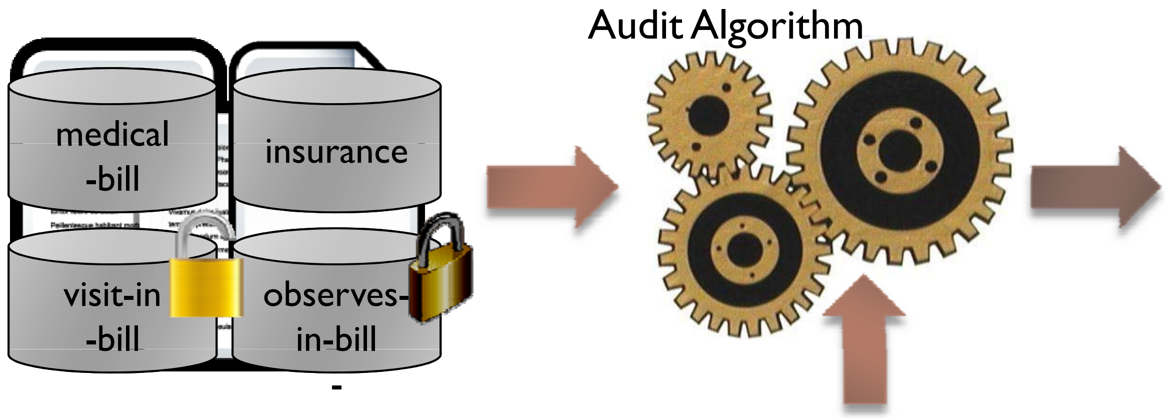
φ'_1
∧ insurance(q,p,c,o,t')
∧ **(visits-in-bill(q,p2,vl,o,t')**
~~∨ observes-in-bill(q,p2,ty,va,o,t')~~))

To
Auditor

Audit
Agent

[8,9]

Example Scenario



φ_2 =
<POL/DISCLOSE>True
(send(Smith, Kosta,D1,date-of-Access) ∧ tagged(D1,Grace)
∧.. ∧ visits-in-bill(Grace,Kosta, O,CARLE,DOB))

***φ’₁* =**
∀p1,p2,m,q,t,ty,va,tp,vl,o,p,c
(send(p1, p2, m, t) ∧ tagged(m, q) ∧ includes(m,ty,va,t)
∧ patientInfo(q,tp,vl,t) ∧ organization(p2,o,t)
∧insuranceInfo(q,p,c,t))
⊃ ∃t’. medical-bill(q,b,t’)
∧ ((time-in(t,t’,t+365)
∧ insurance(q,p,c,o,t’)
∧ **(visits-in-bill(q,p2,vl,o,t’)**
~~∨ observes-in-bill(q,p2,ty,va,o,t’)~~))

Dr. Kosta accessed
Grace’s record on
date-of-access
because Grace
visited Dr. Kosta

[8,9]

The Current State of the Art for Secure Audit Logging Systems ***– Health Care Exchanges***

Oh et al (2014) – Summary

Pro:

- **Solid application of HIBE in Health Care to solve a clear problem**.
- **Preserves privacy** within the Audit Logging system domain.

Cons / Future work:

- **Security appears to only be guaranteed within the Audit Logging system** domain, <u>**not back to the original sources**</u>**.**
- **A need exists to secure against potential alteration of the Audit Log <u>including securing the explanation log</u>**, and the policies used to interrogate the Audit Log.

Future Work

Research such papers as Attila A. Yavuz , Jorge Guajardo, "Dynamic Symmetric **Searchable Encryption** with Minimal Leakage and Efficient Updates on Commodity Hardware" for future applications in health care Audit Logging.

For example, files represent patient records, and attributes/key words can be used for searchable items such as a composite set of addresses, bills due, etc.

Each Patient's EHR

Patient Attributes = bill due, amount, address, etc.

I'	I		…	n
1				
…				
I				
m				

Audit Logging Tools and Systems

Secure Audit Log Tools (e.g., What kind of secure audit log tools are available in the literature?).

The below is a list of Audit Logging Tools:

Number	Product / Company Name	Link:
1	Splunk (Free download/trial)	https://www.splunk.com/en_us/download-5.html
2	AlertLogic Log Manager	https://www.alertlogic.com/solutions/log-correlation-and-analysis/
3	ipswitch (was WhatsUpGold)	https://www.ipswitch.com/solutions/log-and-event-management
4	TIBCO	http://www.tibco.com/products/event-processing/loglogic-for-machine-data
5	GFI EventsManager	http://www.gfi.com/products-and-solutions/network-security-solutions/gfi-eventsmanager
6	SolarWinds Log & Event Manager (LEM)	http://www.solarwinds.com/log-event-manager
7	ManageEngine EventLogAnalyzer	https://www.manageengine.com/products/eventlog/
8	Tripwire	http://www.tripwire.com/
9	NetIQ	https://www.netiq.com/products/sentinel-log-manager/

Updated list of Audit Log tools based on the 2014 article "Top 47 Log Management Tool" at link: https://blog.profitbricks.com/top-47-log-management-tools/ [10]:

Audit Logging Tools and Systems

Number	Product / Company Name	Link:
10	InTrust / Dell Software	https://software.dell.com/products/intrust/
11	Veriato (was SpectorSoft)	http://www.veriato.com/products/veriato-server-manager
12	McAfee Enterprise Log Manager	http://www.mcafee.com/us/products/enterprise-log-manager.aspx
13	LogRhythm	https://logrhythm.com/index.html
14	TNT Software (was ELM Enterprise Manager)	https://tntsoftware.com/
15	Alien Vault	https://www.alienvault.com/solutions/pci-dss-log-management-monitoring
16	Netwrix Auditor	https://www.netwrix.com/event_log_management.html
17	HP / Arcsight ESM	http://www8.hp.com/us/en/software-solutions/arcsight-esm-enterprise-security-management/index.html?#!&!=&tab=TAB1
18	Sumo Logic	https://www.sumologic.com/application/
19	Novell Sentinel Log Manager – Merged with NetIQ, see above.	https://www.netiq.com/products/sentinel-log-manager/
20	Tenable Log Correlation Engine	http://www.tenable.com/products/log-correlation-engine
21	EventTracker	http://www.eventtracker.com/products/log-manager/
22	Konica Minolta Log Management Utility	http://www.biz.konicaminolta.com/solutions/ps_utilities/logmanagement.html
23	Snare – Auditing and Event Log Management	https://www.intersectalliance.com/our-product/
24	Elasticsearch ELK Stack	https://www.elastic.co/products
25	Logscape	http://logscape.com/
26	Sawmill	https://twitter.com/Sawmill
27	Event Sentry	http://www.eventsentry.com/
28	BalaBit syslog-ng	https://www.balabit.com/network-security/syslog-ng

Audit Logging Tools and Systems

Number	Product / Company Name	Link:
29	CorreLog	https://correlog.com/?vsmaid=35
30	Papertrail	https://papertrailapp.com/
31	Assuria Log Manager	http://www.assuria.com/products-new/assuria-log-manager.html
32	Black Stratus - LOGStorm	http://blackstratus.com/enterprise/
33	BeyondTrust - PowerBroker Event Vault for Windows	https://www.beyondtrust.com/products/powerbroker-auditing-security-suite/
34	SemaText Logsene	https://sematext.com/logsene/
35	Kiwi Syslog Server	http://www.kiwisyslog.com/
36	EIQ – Audit Log Management & SIEM	https://www.eiqnetworks.com/solutions/use-cases/audit-log-management-and-siem
37	LOGalyze	http://www.logalyze.com/
38	CloudAccess Log Management	http://www.cloudaccess.com/log-management/
39	Goliath Technologies - MonitorIT Log Management	http://goliathtechnologies.com/performance-monitoring/event-log-management/
40	Check Point - Logging and Status Software Blade featuring SmartLog	https://www.checkpoint.com/products-solutions/security-management/integrated-threat-management/
41	ApexSQL Log	http://www.apexsql.com/sql_tools_log.aspx?utm_source=mssqltips&utm_medium=product_ad&utm_content=log_product&utm_campaign=%5bMSSQL%5d+Log-Product
42	AccelOps Security Information and Event Management (SIEM)	https://www.fortinet.com/products-services/products/siem/fortisiem.html
43	Scalyr	https://www.scalyr.com/?gclid=CODq0sK9t74CFe47MgoddzQAaA
44	Graylog2	https://www.graylog.org/
45	fluentd	http://www.fluentd.org/

Application Videos/ Demo Links

Tripwire:

https://www.demochimp.com/app/view/p/8ffjhbx7

(check as “very important” = “Integrity Monitoring” and “Policy Management”, and check others as “Not Important”)

Splunk:

http://localhost:8000/en-US/manager/search/datainputstats

https://www.splunk.com/en_us/resources/video.UzaWVuNjE60_AMjGA_NfnDfE2FGoIlFB.html

References

Background / Historical References:

[1] Berkeley "Security Audit Logging Guideline" at website link: https://security.berkeley.edu/security-audit-logging-guideline, last visited Sept. 25, 2016.

[2] Karen Kent, Murugiah Souppaya, "Guide to Computer Security Log Management", NIST (National Institute of Standards and Technology) Special Publication 800-92, 2006.

[3] Bruce Schneier, John Kelsey, "Secure Audit Logs to Support Computer Forensics", ACM Transactions on Information and System Security (TISSEC):Volume 2 Issue 2, May 1999.

[4] Scott Crosby, Dan Wallach, "Efficient Data Structures for Tamper-Evident Logging", SSYM'09 Proceedings of the 18th conference on USENIX security symposium Pages 317-334, 2009.

[5] Vipal Goyal et al., "Attribute-based encryption for fine-grained access control of encrypted data", ACM CCS, Proceedings of the 13th ACM conference on Computer and communications security, 2006

Primary References:

[6] Gunnar Hartung, "Secure Audit Logs with Verifiable Excerpt – Full Version", ACM, International Association for Cryptologic Research, 2016 – cites Crosby [4]

[7] Gunnar Hartung, , "Secure Audit Logs with Verifiable Excerpt – Full Version", Presentation Material. KIT – University of the State of Baden-Wuerttemberg and National Laboratory of the Helmholtz Association, 2016

[8] Se Eun Oh, et al., "Privacy-preserving audit for broker-based health information exchange", ACM, Proceedings of the 4th ACM conference on data and application security and privacy, 2014

[9] Se Eun Oh, et al. "Privacy-preserving audit for broker-based health information exchange", Presentation Material, Illinois Security Lab, 2014

Tools References:

[10] Andy Lurie, "Top 47 Log Management Tools", In Cloud Computing, May 19, 2014, at link: , https://blog.profitbricks.com/top-47-log-management-tools/, Last visited Sept 27, 2016.

Questions?

THANK YOU!