GRINGOTTs: Securing Data for Digital Evidence

Catherine M.S. Redfield & Hiroyuki Date
SECOM Co., Ltd.
Intelligent Systems Laboratory
IEEE Data Usage and Management Workshop
May 17, 2014
Outline

• Introduction
• Gringotts Signature Scheme
• Gringotts Storage
• Implementation
• Conclusion
Outline

• Introduction
  • Background
  • Goals
  • Design
• Gringotts Signature Scheme
• Gringotts Storage
• Implementation
• Conclusion
Background

- Technology
  - Business infrastructure is increasingly digital
  - “Paper” trails are harder to follow and can be less trustworthy
- Electronic evidence
  - Not restricted to cybercrime
  - Surveillance camera recordings, server logs, sensor readings, ...

“While some judges believe that their objectivity and precision make it more reliable and therefore they favour its use, others think that the lack of means to verify its authenticity make it more vulnerable and therefore less reliable than traditional evidence.”

-The Admissibility of Electronic Evidence in Court: Fighting Against High-Tech Crime [Cybex06]
Goals

• Usable in conjunction with existing data gathering systems
  • Signature at the data source
    • Disallow altered, added, or deleted packets
    • Non-repudiation: server/database cannot forge data
  • Data usage timing is unknown
  • Practical data transmission speed and bandwidth
• Data stored on the server must be trusted, without having to trust every entity that can access the server
System Design

Data Source

Data Source

Data Source

Server

Database

Verifier

Time Stamp Authority

Data Owner, Court, etc.
Outline

• Introduction
• Gringotts Signature Scheme
  • Related Work
  • Design
• Gringotts Storage
• Implementation
• Conclusion
## Related Work

<table>
<thead>
<tr>
<th>Existing Scheme</th>
<th>Impractically Slow</th>
<th>Lossy</th>
<th>Single Data Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gennaro, Rohatgi ’97</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tartary et al. ’11</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wong, Lam ’98</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Golle, Modadugu ’99</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Perrig et al. ‘00, ’01</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Chu et al. ’00</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Haridasan, Renesse ’06</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

**Digital Signature (SIG)**
Authenticates the sender and the integrity of the message

**Message Authentication Code (MAC)**
Authenticates the integrity of the message
Gringotts Signature Scheme

Source

Data Generation

\[ M_1 = \langle 1, P_1, \text{MAC}(1, P_1) \rangle \]

\[ S_1 = \langle 1, \sigma_1 = \text{SIG}_{\text{sender}}(P_1P_2...P_t), \text{MAC}(1, \sigma_1) \rangle \]

\langle \text{SIG ACK} \rangle

Server

Data Generation

\[ M_2 \]

\[ M_3 \]

\[ M_t \]

\[ M_{t+1} \]

\[ M_{t+2} \]

\[ \text{SIG Verifier} \]

\[ \text{MAC Verifier} \]

\[ \text{Database Storage} \]
Gringotts Signature Scheme

Source

Data Generation

\[ M_1 = <1, P_1, \text{MAC}(1, P_1)> \]

\[ S_1 = <1, \sigma_1 = \text{SIG}_{sender}(P_1 P_2 \ldots P_t), \text{MAC}(1, \sigma_1)> \]

\[ <\text{resend } M_2 \text{ request}> \]

\[ M_2 \]

\[ <\text{SIG ACK}> \]

Server

\[ \text{MAC Verifier} \]

\[ \text{SIG Verifier} \]

\[ \text{Database Storage} \]
Outline

- Introduction
- Gringotts Signature Scheme
- Gringotts Storage
  - Evidence Record Syntax
  - Ordered ERS
  - Gringotts Server
- Implementation
- Conclusion
Evidence Record Syntax (ERS)

- Long-term storage protocol
- A trusted third party acts as a Time Stamp Authority (TSA)
  - Provides certification of data integrity and existence a specific time
- The use of hash trees for storage allows efficient updates when signature or hash algorithms become obsolete
- [ERS07]
Ordered ERS

- ERS extension that also protects file order
- Adds metadata files to the hash tree to build a linked list between sequential hash trees
  - Register (xml/dat)
  - Terminal (xml)
- [Date11]
Outline

- Introduction
- Gringotts Signature Scheme
- Gringotts Storage
- Implementation
  - Test system
  - Signature Algorithms
  - Key Management
  - Performance
- Conclusion
Test System

Data Source → Server → Database → Verifier → Time Stamp Authority

iPad camera:
- 1GHz ARM Cortex A9
- processor 1GB of 1024MHz DDR2 memory
- iOS6

Signature Scheme: C as a custom URLProtocol for ffmpeg

Mac mini:
- 2.5GHz Intel Core i5 processor
- 4GB of 1333MHz DDR3 memory
- OS X 10.8

Database/Verifier: Java using the SECOM ERS library with a custom Ordered ERS implementation

IS Lab experimental TSA
Signature Algorithms

Elliptic Curve Digital Signature Algorithm (ECDSA)
Hash-Based Message Authentication Code (HMAC)
Key Management
Performance: Time

- Data Only
- Gringotts +7.5%

Time (s)

0 0.1 0.2 0.3 0.4 0.5 0.6
Performance: Space

- Ordered ERS (+4.7%)
- ERS Data (+1.1%)
- Signature (+0.38%)
- Packet Size (avg)
Outline

- Introduction
- Gringotts Signature Scheme
- Gringotts Storage
- Implementation
- Conclusion
Conclusion

• Contributions
  • An efficient novel streamed data signature scheme that is not lossy by default
    • Can also be used for streamed log files, etc.
  • Long-term storage system for streamed data with signatures generated by data sources

• Future work:
  • Distributed ERS implementation
  • Comparable data timestamps
  • Secure key distribution and management
Works Cited

- Chu, Rao, Sashan, Zhang. “A Case for End System Multicast.” [Chu00]
- 伊達浩行・佐藤雅史「データの順序関係を検証可能とする、ERS規格のタイムスタンプを付与する電子署名装置」特許申請中。[Date11]
- Gennaro, Rohatgi. “How to Sign Digital Streams.” [Gennaro97]
- Honovich. “How to Design a Video Surveillance Solution.” [Honovich12]
- The IETF Trust. “Request for Comments 4998: Evidence Record Syntax.” [ERS07]
- Lewis. “Every step you take: UK underground centre that is spy capital of the world.” [Lewis09]
- Perrig, Canetti, Tygar, and Song. “Efficient Authentication and Signing of Multicast Streams over Lossy Channels.” [Perrig00]
- Perrig, Canetti, Song, and Tygar. “Efficient and Secure Source Authentication for Multicast.” [Perrig01]
Questions/Comments