Outsourcing the Decryption of ABE Ciphertexts

Matthew Green and Susan Hohenberger
Johns Hopkins University

Brent Waters
UT Austin
Background

• A problem

• Securing records in a data-sharing environment

• E.g., medical records, sensitive documents, etc.

• Share with some but not all

“All cardiologists who work at Johns Hopkins”
Traditional access control

- Relies on a trusted party (reference monitor)

- Non-cryptographic

- Well-known drawbacks: software, insiders, availability
Cryptographic access control

• Traditional approach (public-key encryption)
  • Encrypt record to all valid recipients
  • Problem: must know all possible recipient keys
Cryptographic access control

- Traditional approach (public-key encryption)
  - Encrypt record to all valid recipients
  - Problem: must know all possible recipient keys
  - What if principals change?
ABE

• Attribute-Based Encryption [Sahai-Waters ’05]

• Extension of Identity-Based Encryption

• Encrypt to users with certain attributes

“All cardiologists who work at Johns Hopkins”
CP-ABE

• Ciphertext-policy ABE [BSW07]
  
• User secret keys bound to a list of attributes

• Users obtain keys from an authority

Key Authority
CP-ABE

- Ciphertext-policy ABE [BSW07]

- Encryptors can specify a policy as a boolean formula over attributes

(“Cardiologist” AND “Johns Hopkins”) OR “X-Ray Tech”
CP-ABE

• Ciphertext-policy ABE [BSW07]

• Formulae can use arbitrary numbers of AND, OR, (m-of-n Threshold) gates

((“Cardiologist” AND “Johns Hopkins”) OR “X-Ray Tech”)
CP-ABE

- Ciphertext-policy ABE [BSW07]

- Formulae can use arbitrary numbers of AND, OR, (m-of-n Threshold) gates

- Using these gates we can build <, >, = operators by representing quantities as binary values

((“Cardiologist” AND “Johns Hopkins”) OR “X-Ray Tech”) AND KeyCreationDate > 1313096813
CP-ABE

• Ciphertext-policy ABE [BSW07]

• Formulae can use arbitrary numbers of AND, OR, (m-of-n Threshold) gates

• Using these gates we can build $<, >, =$ operators by representing quantities as binary values

(("Cardiologist" AND "Johns Hopkins") OR "X-Ray Tech") AND KeyCreationDate > 1313096813

This is a 32-element boolean subformula
KP-ABE

• Key-policy ABE [SW05]

• All of the same ideas, but policy/attributes are reversed

• Each ciphertext contains a list of attributes, each key a boolean policy formula

"LabReport", "XRay", "Cardiac"

(\texttt{LabReport AND Cardiac})
\texttt{OR XRay}
So what’s the problem?

• We have this ABE stuff
  • It lets us implement *arbitrarily* complex encryption policies
  • Doesn’t require an on-line reference monitor
  • Why can’t we just use it?
So what’s the problem?

- Two small wrinkles:
  - Ciphertext size and decryption time grow with the complexity of the access policy (resp. attribute list)

Waters09 CP-ABE scheme, 224-bit MNT curve
So what’s the problem?
To the cloud?
Naive Approach
Naive Approach

Remote ciphertext location

Amazon Web Services

Azure
Naive Approach

Remote ciphertext location

Plaintexts (smaller)

Wednesday, August 31, 11
Naive Approach

• Problem:
  • We really need to trust the cloud
  • And every fellow cloud user
    • Timing attacks
    • VM exploits
    • CCA attacks
Other approaches

• Why not generic outsourcing techniques?
  
  • E.g., Craig Gentry’s fully-homomorphic encryption
    
    • This protects the secret key
    
    • Far too inefficient [GH11]
  
  • Outsourcing pairings [CmCMNS10]
    
    • Still costly, high bandwidth
Our Approach

• Change the way that ABE secret keys are generated

• Authority produces a Transform Key and an Elgamal-style Secret Key
Our Approach

- **TK** can go to anyone (e.g., the cloud)
- Client retains **SK**
Our Approach

• Change the way that ABE secret keys are generated

• Also define two new algorithms:
  
  • **Transform**
  
  • **DecOut**

```
Transform(TK, C) -> C'
```

```
DecOut(SK, C')
```

“Partially-decrypted” ciphertext (smaller!)
Outsourcing Security Model

• Traditional CP- (resp. KP-) ABE security def’n:
  • Adversary can query for any secret keys it wants
  • Eventually it asks for a challenge ciphertext on any policy (resp. attr list) not covered by those keys

• New wrinkle:
  • Adversary can query for $TK$ on any policy (resp. attr list) with no restrictions at all (i.e., regardless of the challenge)
  • This models a fully adversarial outsourcing party
Construction: CP-ABE

- Original Waters '09 construction (prime-order bilinear):

\[
\text{MPK} = g, \ e(g, g)^{\alpha}, \ g^a
\]
\[
\text{ABE-SK} = K' = g^\alpha g^{at} \quad L' = g^t \quad \forall x \in S \quad K'_x = H(x)^t.
\]
Construction: CP-ABE

• Original Waters '09 construction:

\[ \text{MPK} = g, \ e(g, g)^\alpha, \ g^\alpha. \]

\[ \text{ABE-SK} = K' = g^\alpha g^{at} \quad L' = g^t \quad \forall x \in S \quad K'_x = H(x)^t. \]

Pick random \( SK = z \) in \( \mathbb{Z}_q \)

\[ TK = K = K'^{1/z} \quad L = L'^{1/z} \quad \{ K_x \}_{x \in S} = \{ K'_x^{1/z} \}_{x \in S} \]
Construction: CP-ABE

• Original Waters '09 construction:

**Encryption:**

\[ C = M \cdot e(g, g)^{\alpha s}, \quad C' = g^s, \]
\[ (C_1 = g^{a\lambda_1} \cdot F(\rho(1))^{-r_1}, \quad D_1 = g^{r_1}), \ldots, \quad (C_\ell = g^{a\lambda_\ell} \cdot F(\rho(\ell))^{-r_\ell}, \quad D_\ell = g^{r_\ell}) \]

**Transform:**

\[ e(C', K)/\left(e(\prod_{i \in I} C_i^{\omega_i}, L) \cdot \prod_{i \in I} e(D_i^{\omega_i}, K_{\rho(i)})\right) = e(g, g)^{s\alpha/z} \]
\[ e(g, g)^{ast} / (\prod_{i \in I} e(g, g)^{ta\lambda_i\omega_i}) = e(g, g)^{s\alpha/z} \]
Construction: CP-ABE

- Original Waters ‘09 construction:

Encryption:

\[
C = M \cdot e(g, g)^{\alpha s}, \quad C' = g^s, \\
(C_1 = g^{\alpha \lambda_1} \cdot F(\rho(1))^{-r_1}, \quad D_1 = g^{r_1}), \ldots, \quad (C_\ell = g^{\alpha \lambda_\ell} \cdot F(\rho(\ell))^{-r_\ell}, \quad D_\ell = g^{r_\ell})
\]

Transform:

\[
e(C', K)/ (e(\prod_{i \in I} C_i^{\omega_i}, L) \cdot \prod_{i \in I} e(D_i^{\omega_i}, K_{\rho(i)})) = e(g, g)^{s\alpha/z} \cdot e(g, g)^{ast} / (\prod_{i \in I} e(g, g)^{t_a \lambda_i \omega_i}) = e(g, g)^{s\alpha/z}
\]
Construction: CP-ABE

- Original Waters '09 construction:

\[
C = \mathcal{M} \cdot e(g, g)^{\alpha s}, \quad C' = g^s, \\
(C_1 = g^{\alpha \lambda_1} \cdot F(\rho(1))^{-r_1}, \quad D_1 = g^{r_1}), \ldots, (C_\ell = g^{\alpha \lambda_\ell} \cdot F(\rho(\ell))^{-r_\ell}, \quad D_\ell = g^{r_\ell})
\]

**Encryption:**

\[
e(C', K)/ (e(\prod_{i \in I} C_i^{\omega_i}, L) \cdot \prod_{i \in I} e(D_i^{\omega_i}, K_{\rho(i)})) = e(g, g)^{s\alpha/z} \cdot e(g, g)^{a_{st}} / (\prod_{i \in I} e(g, g)^{t_{a\lambda_i} \omega_i}) = e(g, g)^{s\alpha/z}
\]

**Transform:**

\[
\text{Transformed ciphertext: } e(g, g)^{s\alpha/z} \cdot \mathcal{M} \cdot e(g, g)^{\alpha_s}
\]
Additional Constructions

• In the paper:
  
  • Security proofs
  
  • An additional scheme from the Goyal et al. Key-policy ABE [GPSW06]
  
  • Also: CCA Security for both CP- and KP-ABE (random oracles)
Performance: Waters09

- 3GHz Intel Core Duo, 4GB RAM (one core)
- 412Mhz ARM (iPhone 3G)

No Outsourcing

![Graph showing ABE Decryption Time]

- Time in seconds
- Number of policy leaves (N)
Performance: Waters09

- 3GHz Intel Core Duo, 4GB RAM (one core)
- 412Mhz ARM (iPhone 3G)

No Outsourcing

With Outsourcing

---

Wednesday, August 31, 11
Ciphertext Size: Waters09

Before Transform

![Graph showing ABE ciphertext size (in bytes) against number of policy attributes (N)]

After Transform

![Graph showing partially-decrypted ciphertext size (in bytes) against number of policy attributes (N)]

<table>
<thead>
<tr>
<th>ABE</th>
<th>Elgamal</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABECiphertext Size</td>
<td>Ciphertext Size</td>
</tr>
</tbody>
</table>
An EC2-based System

- We constructed Amazon Machine Image ("Proxy") with:
  - Apache
  - Scripts to accept a Transform Key, load ciphertexts from remote URLs
  - The code for the Transform algorithm
  - Users can programmatically spin up one or more instances
An EC2-based System

• Also created a test application

• Extended the *iHealthEMR* app from JHU (Ayo Akinyele) (Medical records reader, uses CP-ABE)

• Added code to transparently instantiate Proxy, upload Transform Key at startup

• 1-1.5 min for spinup, during which decryption is local.

• Afterwards it’s outsourced!
## An EC2-based System

<table>
<thead>
<tr>
<th>Operation</th>
<th>local-only (sec)</th>
<th>local+web (sec/kb)</th>
<th>proxy (sec/kb)</th>
<th>proxy+web (sec/kb)</th>
</tr>
</thead>
<tbody>
<tr>
<td>New proxy instantiation</td>
<td>·</td>
<td>·</td>
<td>93.4 sec</td>
<td>93.4 sec</td>
</tr>
<tr>
<td>Restart existing proxy instance</td>
<td>·</td>
<td>·</td>
<td>45 sec</td>
<td>45 sec</td>
</tr>
<tr>
<td>Generate &amp; set 70-element transform key</td>
<td>·</td>
<td>·</td>
<td>2.9 sec</td>
<td>2.9 sec</td>
</tr>
<tr>
<td>Decryption:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>((DOCTOR OR NURSE) AND INSTITUTION)</td>
<td>1.1s</td>
<td>1.2s/1.1k</td>
<td>.2s/1.4k</td>
<td>.2s/0.4k</td>
</tr>
<tr>
<td>(DOCTOR AND TIME &gt; 1262325600 AND TIME &lt; 1267423200)</td>
<td>17.3s</td>
<td><strong>17.3s/22.8k</strong></td>
<td>1.2s/23.2k</td>
<td><strong>1.2s/0.4k</strong></td>
</tr>
</tbody>
</table>
Other Applications

• Outsourcing from smartcards
  • Let the computer do the heavy lifting!
  • Simplify the code base on the smart card
• Reducing TCB
  • ABE implementations are complex: parsing code, excess cryptography == vulnerabilities?
  • Let’s not trust that all that code:
    • Isolate one trusted piece using e.g., TrustVisor [MLQZDGP10].
Open Problems

• Outsourcing for other cryptosystems (IBE, ABE, NIZKs, Signatures)

• CCA security in the standard model

• A generic cloud-based outsourcing platform
  • Supports many cryptosystems
  • Attacker uploads code of his/her choice at initialization time