Two Secure Anonymous Proxy-based Data Storages

Olivier Blazy\textsuperscript{1} \hspace{1cm} Xavier Bultel\textsuperscript{2} \hspace{1cm} Pascal Lafourcade\textsuperscript{2}

Université de Limoges, Xlim, Limoges, France

Clermont Université Auvergne, LIMOS, Clermont-Ferrand, France

July 29, 2016

SECRYPT 2016, Lisbon

\*This research was conducted with the support of the “Digital Trust” Chair from the University of Auvergne Foundation.
Proxy Re-Encryption (PRE)

Alice ($pk_a, sk_a$)

Bob ($pk_b, sk_b$)

Proxy

Plearns nothing about $m$ (IND-CPA).
Proxy Re-Encryption (PRE)

Alice \((pk_a, sk_a)\)  

Bob \((pk_b, sk_b)\)

Proxy

re-key \(rk_{b \rightarrow a}\)

\[ c' = RE_{rk_{b \rightarrow a}}(c) = E_{pk_a}(m) \]

Plearns nothing about \(m\) (IND-CPA).

Olivier Blazy, Xavier Bultel, Pascal Lafourcade (Université de Limoges, Xlim, Limoges, France, Clermont Université Auvergne, LIMOS, Clermont-Ferrand, France)

Two Secure Anonymous Proxy-based Data Storages

July 29, 2016 2 / 20
Proxy Re-Encryption (PRE)

Alice \((pk_a, sk_a)\)

(rk\(_{b\rightarrow a}\))

Bob Offline

\[
c = E_{pk_b}(m) \quad c' = \text{RE}_{rk_{b\rightarrow a}}(c) = E_{pk_a}(m) \quad m = D_{sk_a}(c')
\]

Plearns nothing about \(m\) (IND-CPA).
Proxy Re-Encryption (PRE)

Alice $(pk_a, sk_a)$

Bob Offline

Proxy

$c = E_{pk_b}(m)$

$c' = RE_{rk_{b\rightarrow a}(c)} = E_{pk_a}(m)$

Plearns nothing about $m$ (IND-CPA).

Olivier Blazy, Xavier Bultel, Pascal Lafourcade (Université de Limoges, Xlim, Limoges, France, Clermont Université Auvergne, LIMOS, Clermont-Ferrand, France)
Proxy Re-Encryption (PRE)

Alice \((pk_a, sk_a)\) → Proxy \((rk_{b\rightarrow a})\)

\[ c = E_{pk_b}(m) \]

\[ c' = RE_{rk_{b\rightarrow a}}(c) = E_{pk_a}(m) \]

Bob Offline

\[ c' = E_{pk_a}(m) \]

Plearns nothing about \(m\) (IND-CPA).
Proxy Re-Encryption (PRE)

Alice \((pk_a, sk_a)\)

Bob Offline

\(c = E_{pk_b}(m)\)

\(m = D_{sk_a}(c')\)

\(c' = RE_{rk_{b\to a}}(c)\)

\(c' = E_{pk_a}(m)\)

Olivier Blazy, Xavier Bultel, Pascal Lafourcade (Université de Limoges, Xlim, Limoges, France, Clermont Université Auvergne, LIMOS, Clermont-Ferrand, France)

Two Secure Anonymous Proxy-based Data Storages

July 29, 2016 2 / 20
Proxy Re-Encryption (PRE)

Alice $(pk_a, sk_a)$

$c = E_{pk_b}(m)$

$m = D_{sk_a}(c')$

$\text{Proxy}$

$c' = RE_{rk_{b\to a}}(c)$

$c' = E_{pk_a}(m)$

Bob Offline

$P$ learns nothing about $m$ (IND-CPA).
PRE History

- Blaze et al. (1998) First definition of PRE.
- Ateniese et al. (2006) Unidirectional PRE.
- Canetti et al. (2007) CCA security.
- Libert et al. (2007) Unidirectional + CCA.
PRE History

- Blaze et al. (1998) First definition of PRE.
- Ateniese et al. (2006) Unidirectional PRE. **New application: encrypted storage management.**
- Canetti et al. (2007) CCA security.
- Libert et al. (2007) Unidirectional + CCA.
PRE based storage

Owner \((pk_o, sk_o)\)

User \((pk_u, sk_u)\)

Encrypted storage

Semi-trust proxy:
- No info about \(m\) - \(P\) knows \(U\) id.
- \(P\) knows \(U\) rights
- \(P\) knows \(c\)

Goal: more privacy!

Olivier Blazy, Xavier Bultel, Pascal Lafourcade (Université de Limoges, Xlim, Limoges, France, Clermont Université Auvergne, LIMOS, Clermont-Ferrand, France)
PRE based storage

Owner \( (pk_o, sk_o) \)

User \( (pk_u, sk_u) \)

Proxy

Encrypted storage

re-key \( rk_{o\rightarrow u} \)

Check user rights

C = \( E_{pk_o}(m) \)

\( c' = E_{pk_u}(m) \)

\( m = D_{sk_u}(c') \)

Semi-trust proxy:
- No info about \( m \)
- \( P \) knows \( U \) id.
- \( P \) knows \( U \) rights
- \( P \) knows \( c \)

Goal: more privacy!

Olivier Blazy, Xavier Bultel, Pascal Lafourcade (Université de Limoges, Xlim, Limoges, France, Clermont Université Auvergne, LIMOS, Clermont-Ferrand, France)

Two Secure Anonymous Proxy-based Data Storages

July 29, 2016 4 / 20
PRE based storage

User \((pk_u, sk_u)\)

Owner Offline

Encrypted storage

Semi-trust proxy:
- No info about \(m\) - \(P\) knows \(U\) id.
- \(P\) knows \(U\) rights
- \(P\) knows \(c\)

Goal: more privacy!

Olivier Blazy, Xavier Bultel, Pascal Lafourcade
Two Secure Anonymous Proxy-based Data Storages
July 29, 2016 4 / 20
PRE based storage

User \((pk_u, sk_u)\)

Owner Offline

\(rk_{o\rightarrow u}\)

Check user rights

file \(c\) ?

Proxy

Encrypted storage

file \(c\) ?

PRE based storage

User \((pk_u, sk_u)\)

Owner Offline

\[ c' = E_{pk_u}(m) \]

Proxy

\[ m = D_{sk_u}(c') \]

\[ c' = RE_{rk_{o \rightarrow u}}(c) \]

Encrypted storage

Goal: more privacy!

Olivier Blazy, Xavier Bultel, Pascal Lafourcade (Université de Limoges, Xlim, Limoges, France, Clermont Université Auvergne, LIMOS, Clermont-Ferrand, France)
Semi-trust proxy:
- No info about $m$
- $P$ knows $U$ id.
- $P$ knows $U$ rights
- $P$ knows $c$

User ($pk_u, sk_u$)

$c' = E_{pk_u}(m)$

$m = D_{sk_u}(c')$

$rk_{o \rightarrow u}$

$c = E_{pk_o}(m)$

$c' = RE_{rk_{o \rightarrow u}}(c)$

Encrypted storage
PRE based storage

User \((pk_u, sk_u)\)

Owner Offline

Semi-trust proxy:
- No info about \(m\)
- \(P\) knows \(U\) id.
- \(P\) knows \(U\) rights
- \(P\) knows \(c\)

Goal: more privacy!

\[ c' = E_{pk_u}(m) \]
\[ m = D_{sk_u}(c') \]

\[ c = E_{pk_o}(m) \]
\[ c' = RE_{rk_o \rightarrow u}(c) \]
PRE & anonymity?

- Zheng et al. (2014) Anonymous re-encryption key + CCA.
PRE & anonymity?

- Zheng et al. (2014) Anonymous re-encryption key + CCA.

→ Only partial anonymity.
Our idea

Owner \((pkg_i, skg_i)\)

User

member of the group \(i\)
Our idea

Owner \((\text{pkg}_i, \text{skg}_i)\)

proxy key \(K\)

Member key \(\text{MSK}_i\)

User

member of the group \(i\)

Encrypted storage
Our idea

Owner Offline

\[ K \]

User (MSK\(_i\))

Encrypted storage

\[ c = E_{pkg_i}(m) \]

User knows MSK\(_i\) and \(c\)

Randomization with \(r\) MSK\(_i\)' and \(c\) MSK\(_i\)'

\[ c' = R_{K, MSK_i}(c') \]

\[ m = D_{r}(c') \]

Semi-trust proxy:
- No info about \(m\)
- No info about User
- No info about \(i\)
- No info about \(c\)
Our idea

Owner Offline

User (MSKₖ)

K

Proxy

file c?

Encrypted storage

User knows MSKₖ and c

Randomization with r

MSKₖ′ and cₖ′

\[ cₖ′ = \text{RE}_{K_{MSKₖ'}}(cₖ) \]

\[ m = \text{DR}(cₖ′) \]

Semi-trust proxy:
- No info about m
- No info about User
- No info about K
- No info about c
Our idea

Owner Offline

User (MSK<sub>i</sub>)

\[ c = E_{pkg_i}(m) \]

Encrypted storage

\[ K \]

Proxy

User knows MSK<sub>i</sub> and <i>c</i>
Randomization with r<sub>MSK<sup>′</sup><sub>i</sub></sub> and <i>c</i><sup>′</sup><sub>MSK<sup>′</sup><sub>i</sub></sub>, <i>c</i><sup>′′</sup><sub>MSK<sup>′</sup><sub>i</sub></sub>, <i>c</i><sup>′′</sup><sub>MSK<sup>′</sup><sub>i</sub></sub>

Semi-trust proxy:
- No info about \( m \)
- No info about User
- No info about i
- No info about c
Our idea

Owner Offline

\[ c = E_{\text{pkg}_i}(m) \]

User knows
MSK\(_i\) and \( c \)

User (MSK\(_i\))

Encrypted storage

Randomization with \( r \), \( \text{MSK}'_i \) and \( c' \)
\[ c'' = \text{RE}_{\text{K}, \text{MSK}'_i}(c') \]
\[ m = \text{DR}_{\text{MSK}'_i}(c'') \]

Member key MSK\(_i\)
Proxy key \( K \)

Semi-trust proxy:

- No info about \( m \)
- No info about User
- No info about Owner
- No info about \( i \)
- No info about \( c \)
Our idea

Owner Offline

User (MSK$_i$)

Randomization with $r$
MSK'$_i$ and c'

Encrypted storage
Our idea

Owner Offline

Randomization with $r$
$\text{MSK}_i'$ and $c'$

$K$

User ($\text{MSK}_i$)

Proxy

Encrypted storage

$m = \text{D}_{\text{MSK}_i'}(c')$ $\text{MSK}_i'$ and $c'$

$c'' = \text{RE}_{\text{K}, \text{MSK}_i'}(c')$
Our idea

Owner Offline

User (MSK_i)

K

Proxy

Encrypted storage

\[ c'' = RE_{K, MSK'_i}(c') \]

Randomization with \( r \) MSK'_i and c'

Olivier Blazy, Xavier Bultel, Pascal Lafourcade (Université de Limoges, Xlim, Limoges, France, Clermont Université Auvergne, LIMOS, Clermont-Ferrand, France)
Our idea

Owner Offline

Proxy

User (MSK_i)

Encrypted storage

$c'' = RE_{K, MSK'_i}(c')$

$m = D_r(c'')$
Our idea

Semi-trust proxy:
- No info about $m$
- No info about $U$ id.
- No info about $i$
- No info about $c$

Owner Offline

User (MSK$_i$)

$K$

MSK'$_i$ and $c'$

Encrypted storage

$m = D_r(c'')$

Olivier Blazy, Xavier Bultel, Pascal Lafourcade
Our contribution

Two schemes:

- **DRAS**: Direct revocation mechanism. The owner can revoke anybody anytime. Pay-per-download model. Weak anonymity.

- **IRAS**: Indirect revocation mechanism. The owner can revoke users periodically. Monthly-fee model. Full anonymity.
Our contribution

Two schemes:

**DRAS**: Direct revocation mechanism: The owner can revoke anybody anytime.
- Pay-per-download model.
- Weak anonymity.

**IRAS**: Indirect revocation mechanism: the owner can revoke users periodically.
- Monthly-fee model.
- Full anonymity.
Our contribution

Two schemes:

**DRAS:** Direct revocation mechanism: The owner can revoke anybody anytime.
- Pay-per-download model.
- Weak anonymity.

**IRAS:** Indirect revocation mechanism: the owner can revoke users periodically.
- Monthly-fee model.
- Full anonymity.
introduction

DRAS

IRAS

Conclusion
DRAS

P-Gen(\mathcal{P}): generate proxy keys (PKP, SKP).
G-Gen(\mathcal{P}): generate group key (PKG_j, SKG_j).
Join(SKG_j, WL, U_i): generate a group member key MSK^i_j.
Encrypt(PKG_j, m): encrypt m for group j.
Revoke(MSK^i_j, BL): revoke a user.
Open(VIEW, WL): desanonymize a transaction.
ProxyDec(U_i, P): decryption protocol between a user and the proxy.
Keys construction:

- Proxy keys \((PKP, SKP)\) for an encryption scheme.
- Group key \((PKG, SKG) = (g^\gamma, \gamma)\).
- Member key \(MSK = (t, Enc_{PKP}(\frac{t}{\gamma}))\).
DRAS

Keys construction:

- Proxy keys (PKP, SKP) for an encryption scheme.
- Group key (PKG, SKG) = \((g^\gamma, \gamma)\).
- Member key MSK = \((t, \text{Enc}_{PKP}(\frac{t}{\gamma}))\).

The encryption algorithm is an ElGamal variant:

- **Keys** Secret sk = \(x\), public pk = \(g^x\).
- **Encryption** Pick \(r\) and compute \((C_1, C_2) = (pk^r, g^r \cdot m)\).
- **Decryption** Compute \(m = \frac{C_2}{C_1^{1/sk}}\)
DRAS: Decryption protocol

- $C = (C_1, C_2) = (g^{r \cdot \gamma}, g^r \cdot m)$
- $\text{MSK} = (\text{MSK}_1, \text{MSK}_2) = (t, \text{Enc}_{\text{PKP}}(\frac{t}{\gamma}))$
DRAS: Decryption protocol

- \( C = (C_1, C_2) = (g^{r \cdot \gamma}, g^r \cdot m) \)
- \( \text{MSK} = (\text{MSK}_1, \text{MSK}_2) = (t, \text{Enc}_{\text{PKP}}(\frac{t}{\gamma})) \)

\[
\begin{align*}
\text{(PKP; MSK; C)} & \quad \text{\textbf{\rightarrow}} \quad \text{(SKP; BL)} \\
\text{\textbf{s \leftarrow \{Z_p^*\}; B = (C_1)^s}} & \quad \text{\textbf{\rightarrow}} \quad \text{If MSK}_2 \in \text{BL} \text{ then abort; else } w = \text{Dec}_{\text{SKP}}(\text{MSK}_2) \\
\text{\textbf{m = } g^{r \cdot m}} & \quad \text{\textbf{\leftarrow}} \quad \text{D} \\
\text{\textbf{= } g^{s \cdot r \cdot t \cdot \frac{1}{s \cdot t}}} & \quad \text{\textbf{\leftarrow \rightarrow} \quad D = (B)^w = (C_1^s)^{\frac{t}{\gamma}} = g^{s \cdot r \cdot t}} \\
\text{Output m} & \quad \text{Output VIEW = MSK}_2.
\end{align*}
\]
DRAS: Decryption protocol

- $C = (C_1, C_2) = (g^{r \cdot \gamma}, g^r \cdot m)$
- $\text{MSK} = (\text{MSK}_1, \text{MSK}_2) = (t, \text{Enc}_{\text{PKP}}(\frac{t}{\gamma}))$

$\text{(PKP; MSK; C)} \quad \text{(SKP; BL)}$

$s \leftarrow \mathbb{Z}_p^*; B = (C_1)^s \quad \text{If MSK}_2 \in \text{BL then abort;}
\quad \text{else } w = \text{Dec}_{\text{SKP}}(\text{MSK}_2) = \frac{t}{\gamma}$

$m = \frac{C_2}{D^{(1/s \cdot t)}} \quad \leftarrow D
\quad = \frac{g^r \cdot m}{g^{s \cdot r \cdot \frac{1}{s} \cdot t}} = \frac{g^r \cdot m}{g^r}$

Output $m$

Proxy links user who uses two times the same member key

Output $\text{VIEW} = \text{MSK}_2$. 
ElGamal is malleable

- \( C = (g^r, g^{x \cdot r} \cdot m) \)
- \( C' = ((g^r)^s, (g^{x \cdot r} \cdot m)^s) = (g^{r \cdot s}, g^{(r \cdot s) \cdot x} \cdot m^s) \).
- Decryption: \( m' = m^s = \frac{g^{(r \cdot s) \cdot x} \cdot (m^s)}{g^{r \cdot s}} \).
- Difficult to link \( C \) and \( C' \) (Diffie-Hellman problem).
- The message \( m \) is hidden.
Indirect Revocation Anonymous Storage

O-Gen(\(P\)): generate owner (PKO, SKO).
P-Gen(\(P\)): generate proxy key pair (PKP, SKP).
G-Gen(\(P\)): generate group key pair (PKG, SKG).
Join(SKG\(_j\), SKO, PKP): generate a group member key MSK.
O-Update(SKO, PKO): update (PKO, SKO).
U-Update(MSK\(_j\), SKO): update MSK.
Encrypt(PKG\(_j\), m): encrypt a message \(m\).
ProxyDec(\(U_i\), \(P\)): decryption protocol between a user and the proxy.
(Simplified) IRAS parameters

Keys constructions:

- $\mathcal{P} = (G_1, G_2, G_T, g_1, g_2, e, \text{PKE}, S)$.
- Proxy keys $(\text{PKP}, \text{SKP}) = (g_2^p, p)$.
- Group keys $(\text{PKG}, \text{SKG}) = (g_1^\gamma, \gamma)$.
- Member key $\text{MSK} = (g_2^{p\cdot s}, g_2^s \cdot g_2^{1/\gamma})$. 
(Simplified) IRAS parameters

Keys constructions:
- $\mathcal{P} = (G_1, G_2, G_T, g_1, g_2, e, PKE, S)$.
- Proxy keys $(PKP, SKP) = (g_2^p, p)$.
- Group keys $(PKG, SKG) = (g_1^\gamma, \gamma)$.
- Member key $MSK = (g_2^{p\cdot s}, g_2^s \cdot g_2^{1/\gamma})$.

The encryption algorithm is an ElGamal bilinear variant:

**Keys** Secret $sk = x$, public $pk = g_1^x$.

**Encryption** Pick $r$ and compute $(C_1, C_2) = (pk^r, e(g_1, g_2)^r \cdot m)$.

**Decryption** Compute $m = \frac{C_2}{e(C_1, g_2)^{1/sk}}$.
(Simplified) Decryption protocol

- \( C = (C_1, C_2) = (g_1^{r \cdot \gamma}, e(g_1, g_2)^r \cdot m) \)
- \( \text{MSK} = (\text{MSK}_1, \text{MSK}_2) = (g_2^{p \cdot s}, g_2^s \cdot g_2^{\frac{1}{\gamma}}) \)
(Simplified) Decryption protocol

- \( C = (C_1, C_2) = (g_1^{r \cdot \gamma}, e(g_1, g_2)^r \cdot m) \)
- \( \text{MSK} = (\text{MSK}_1, \text{MSK}_2) = (g_2^{p \cdot s}, g_2^s \cdot g_2^{-\gamma}) \)

\[\begin{align*}
\alpha, \beta & \leftarrow \mathbb{Z}_p^* \\
\text{MSK}' &= (\text{MSK}_1^\alpha, \text{MSK}_2^\alpha) \\
C'_1 &= C_1^\beta \\
\frac{C_2^1}{D^\alpha \cdot \beta} &= \frac{e(g_1, g_2)^{r \cdot m}}{e(g_1, g_2)^{r \cdot \alpha \cdot \beta}} \\
D &= e(g_1^{r \cdot \gamma \cdot \beta}, \frac{\text{MSK}_2'}{\text{MSK}_1^{1/p}}) \\
\end{align*}\]

\[D = e(g_1^{r \cdot \gamma \cdot \beta}, \frac{\text{MSK}_2'}{\text{MSK}_1^{1/p}}) = e(g_1, g_2)^{r \cdot \alpha \cdot \beta}\]
Provable IRAS

Many tools to construct a provable scheme:

- Proof of a signature on MSK from MSK'.
- Revocation: The owner updates his signing key, but does not re-sign MSK.
- Damgård-ElGamal (CCA1).
- Smooth projective hash functions.
introduction

DRAS

IRAS

Conclusion
DRAS
- Direct revocation.
- Simple and efficient scheme.
- CPA secure.
- Not fully anonymous.

IRAS
- Indirect revocation.
- Not efficient, use complex tools to be provably secure.
- CPA secure.
- Fully anonymous.

Future work
- Increase and simplify IRAS.
- Without Damgård-ElGamal and SPHF.
Thank you for your attention.

Questions?