A Fully Secure Unidirectional and Multi-use Proxy Re-encryption Scheme

Hongbing Wang and Zhenfu Cao*
Department of Computer Science and Engineering, Shanghai Jiao Tong University
No. 800, Dongchuan Road, Shanghai, 200240, P. R. China
flora_wang,zfcao{@sjtu.edu.cn}

ABSTRACT

In a proxy re-encryption scheme, a ciphertext for Alice can be converted into a ciphertext for Bob with the help of a semi-trusted proxy, while the proxy gets no information about the messages encrypted under either key during the conversion. Proxy re-encryption has become more and more popular these years due to its practical applications. In this paper, we propose a fully secure proxy re-encryption scheme with several useful properties, namely, unidirectionality, multi-use, collusion-safe, non-interactivity, and non-transitivity. Our scheme solves one of the six open problems presented by Canetti and Hohenberger at ACM CCS 2007.

General Terms

Security, Algorithms, Theory.

Keywords

Proxy re-encryption, Unidirectional, Multi-use, Bilinear pairing, Standard model, Chosen ciphertext security.

1. INTRODUCTION

Before the concept of the proxy re-encryption (PRE) was brought out, the ciphertext conversion between different users was accomplished by first decrypting the message, then encrypting it with the new key, which implies access to the plaintext and a reliable copy of the new encryption key.

In 1998, Blaze, Bleumer, and Strauss [1] introduced the concept of (atomic) proxy cryptography, with a concrete PRE scheme based on ElGamal public key scheme [5]. In a PRE scheme, a proxy can convert a ciphertext under Alice's (delegator) public key into one intended to Bob (delegatee) with the same plaintext by using a re-encryption key, which is, in general, generated by Alice. Some basic requirements to a PRE scheme are: (1) the proxy should not be able to learn the plaintext during the transformation; (2) no useful information on the secret keys of Alice and Bob can be deduced from the re-encryption key.

	CH [4]	LV [9]	ABH [7]	Our Scheme
Unidirectionality	No	Yes	Yes	Yes
multi-use	Yes	No	No	Yes
collusion-safe	No	Yes	Yes	Yes
Non-interactive	No	Yes	Yes	Yes
Non-transitive	No	Yes	Yes	Yes
Key-private	No	No	Yes	No
Security	CCA	CCA	CPA	CCA
Assumption	DBDH	3QDBDH	eEDBDH	DBDH

Table 1: Comparison Between Some PRE Schemes

For clarity, we adopt Green and Ateniese's notion of "encryption level" [8] to refer to different ciphertexts in the (multi-use) scheme. A ciphertext generated directly using the encrypt algorithm is termed a "*first-level*" ciphertext. The application of the re-encryption algorithm to an i^{th}-*level* ciphertext results in an $(i+1)^{th}$-*level* ciphertext.

1.1 Our Contribution

In this paper, we construct a PRE scheme which gives a solution to the second open problem left by Canetti and Hohenberger in [4], namely, we propose a unidirectional and multi-use IND-CCA2 secure PRE scheme in the standard model. Moreover, our PRE scheme has more useful properties, such as collusion-safe, non-interactivity, and non-transitivity. The security of our PRE scheme is based on the standard DBDH assumption. We briefly compare our scheme to some PRE schemes in Table 1.

2. NEW CONSTRUCTION OF PRE

We use Canetti-Halevi-Katz's technique [3] (using a strongly unforgeable one-time signature) to achieve public verifiability for $l^{th}-level$ $(l > 1)$ ciphertexts. Such a signature consists of a triple of algorithms $Sig = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ such that, on input a security parameter k, $\mathcal{G}$ generates a one-time key pair (svk, ssk), while for any message M, $\mathcal{V}(svk, \sigma, M)$ outputs 1 whenever $\sigma = \mathcal{S}(ssk, M)$ and 0 otherwise [9].

Scheme Description.

Let 1^k be the security parameter, $(q, g, \mathbb{G}_1, \mathbb{G}_T, e)$ be generated by a bilinear group generator on input (1^k), and $Sig = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ be a strongly unforgeable signature scheme. Let g_1, h_1, h_2 and h_3 be four random elements in $\mathbb{G}_1 \setminus \{g\}$. Further, let H_1: $\{0,1\}^* \to \mathbb{Z}_q^*$ and H_2: $\mathbb{G}_T \to \mathbb{G}_1$ be two one-way, collision-resistant cryptographic hash functions. The public parameters are: $par = (q, g, g_1, h_1, h_2, h_3, \mathbb{G}_1, \mathbb{G}_T, e, Sig, H_1, H_2)$. Our PRE scheme consists of the following five algorithms (`KeyGen`, `Enc`, `RKGen`, `ReEnc`, `Dec`):

- `KeyGen`$(par) \to (pk, sk)$: On input par, select $x \in_R \mathbb{Z}_q^*$. Set $pk = g^x$ and $sk = x$.

- $\texttt{Enc}(par, pk, m) \rightarrow C^{(1)}$: To encrypt a message $m \in \mathbb{G}_T$ under pk, select $r \leftarrow_R \mathbb{Z}_q^*$, then compute $C^{(1)} = (c_{1,1}, c_{1,2}, c_{1,3})$, where $c_{1,1} = g^r$, $c_{1,2} = m \cdot e(g_1, pk)^r$, $c_{1,3} = (h_1^{H_1(c_{1,1})} h_2^{H_1(c_{1,1} \| c_{1,2})} h_3)^r$. Finally, output the *first-level* ciphertext $C^{(1)}$.

- $\texttt{RKGen}(par, sk_i, pk_j) \rightarrow rk_{i \rightarrow j}$ $(i \neq j)$: To generate a re-encryption key from pk_i to pk_j for pk_i's proxy P_i, do the following:

  1. Select $r_i \leftarrow_R \mathbb{Z}_q^*$, $K_i \leftarrow_R \mathbb{G}_T$.
  2. Compute $R_1^{(i)} = g^{r_i}$, $R_2^{(i)} = K_i \cdot e(g_1, pk_j)^{r_i}$, $R_3^{(i)} = svk_{P_i}$, $R_4^{(i)} = (h_1^{H_1(R_1^{(i)})} h_2^{H_1(R_1^{(i)} \| R_2^{(i)} \| R_3^{(i)})})^{r_i}$, $R_5^{(i)} = H_2(K_i) \cdot g_1^{-x_i}$, where svk_{P_i} is a publicly available verification key of pk_i's proxy P_i.
  3. Output $rk_{i \rightarrow j} = (R_1^{(i)}, R_2^{(i)}, R_3^{(i)}, R_4^{(i)}, R_5^{(i)})$. The re-encryption key is sent to P_i via a secure channel.

- $\texttt{ReEnc}(par, rk_{i \rightarrow j}, C_i^{(l)}) \rightarrow \{C_j^{(l+1)}, \perp\}$ $(i \neq j, l \geq 1)$:

  1. To re-encrypt a *first-level* ciphertext $C_i^{(l)}$, denoted by $C_i^{(1)}$, do:
     (a) Parse $C_i^{(1)}$ as $(c_{1,1}, c_{1,2}, c_{1,3})$, and $rk_{i \rightarrow j}$ as $(R_1^{(i)}, R_2^{(i)}, R_3^{(i)}, R_4^{(i)}, R_5^{(i)})$.
     (b) Check if $e(g, c_{1,3}) = e(c_{1,1}, h_1^{H_1(c_{1,1})} h_2^{H_1(c_{1,1} \| c_{1,2})} h_3)$ and $e(g, R_4^{(i)}) = e(R_1^{(i)}, h_1^{H_1(R_1^{(i)})} h_2^{H_1(R_1^{(i)} \| R_2^{(i)} \| R_3^{(i)})})$ hold. If either of them fails, return $\perp$. Otherwise, do the following:
     (c) Compute $C = (c'_{1,1}, c'_{1,2}, c'_{1,3}, c'_{2,1}, c'_{2,2}, c'_{2,3}, c'_{2,4})$, where $c'_{1,1} = c_{1,1}$, $c'_{1,2} = c_{1,2} \cdot e(c_{1,1}, R_5^{(i)})$, $c'_{1,3} = c_{1,3}$, $c'_{2,1} = R_1^{(i)}$, $c'_{2,2} = R_2^{(i)}$, $c'_{2,3} = R_3^{(i)}$, $c'_{2,4} = R_4^{(i)}$.
     (d) Let P_i be pk_i's proxy, and ssk_{P_i} be the signing key of P_i corresponding to P_i's verification key $R_3^{(i)}$.
     (e) Run the signing algorithm $\mathcal{S}(ssk_{P_i}, (c'_{1,1}, c'_{1,2}, c'_{1,3}, c'_{2,1}, c'_{2,3}, c'_{2,4}))$ to generate a signature on the ciphertext tuple $(c'_{1,1}, c'_{1,2}, c'_{1,3}, c'_{2,1}, c'_{2,3}, c'_{2,4})$, and denote the signature as $S_i^{(1)}$.
     (f) Output the ciphertext $C_j^{(2)} = \langle C, S_i^{(1)} \rangle$.
  2. To re-encrypt an l^{th}-*level* $(l > 1)$ ciphertext $C_i^{(l)}$,
     (a) Parse $C_i^{(l)}$ as $(c_{1,1}, \cdots, c_{l,1}, c_{l,2}, c_{l,3}, c_{l,4}, c_{l,5})$, and $rk_{i \rightarrow j}$ as $(R_1^{(i)}, R_2^{(i)}, R_3^{(i)}, R_4^{(i)}, R_5^{(i)})$.
     (b) Check if $e(g, c_{l,4}) = e(c_{l,1}, h_1^{H_1(c_{l,1})} h_2^{H_1(c_{l,1} \| c_{l,2} \| c_{l,3})})$ and $e(g, R_4^{(i)}) = e(R_1^{(i)}, h_1^{H_1(R_1^{(i)})} h_2^{H_1(R_1^{(i)} \| R_2^{(i)} \| R_3^{(i)})})$ hold. If either of them fails, return $\perp$.
     (c) For $\forall\ k \in [2, l]$, check $\mathcal{V}(c_{k,3}, c_{k,5}, (c_{1,1}, \cdots, c_{k,1}, c_{k,3}, c_{k,4})) = 1$.
     Whenever one of them fails, return $\perp$. Otherwise, do the following:
     (d) Compute $C = (c'_{1,1}, \cdots, c'_{l,1}, c'_{l,2}, c'_{l,3}, c'_{l,4}, c'_{l,5}, c'_{l+1,1}, c'_{l+1,2}, c'_{l+1,3}, c'_{l+1,4})$, where $c'_{l,2} = c_{l,2} \cdot e(c_{l,1}, R_5^{(i)})$, $c'_{l+1,1} = R_1^{(i)}$, $c'_{l+1,2} = R_2^{(i)}$, $c'_{l+1,3} = R_3^{(i)}$, $c'_{l+1,4} = R_4^{(i)}$, and all other elements remain unchanged.
     (e) Let P_i be pk_i's proxy, and ssk_{P_i} be the signing key of P_i corresponding to P_i's verification key $R_3^{(i)}$.
     (f) Run the signing algorithm $\mathcal{S}(ssk_{P_i}, (c'_{1,1}, \cdots, c'_{l+1,1}, c'_{l+1,3}, c'_{l+1,4}))$ to generate a signature on the ciphertext tuple $(c'_{1,1}, \cdots, c'_{l+1,1}, c'_{l+1,3}, c'_{l+1,4})$, and denote the signature as $S_i^{(l)}$.
     (g) Output the ciphertext $C_j^{(l+1)} = \langle C, S_i^{(l)} \rangle$.

- $\texttt{Dec}(par, sk_i, C_i^{(l)}) \rightarrow \{m, \perp\}$ $(l \geq 1)$: If $C_i^{(l)}$ can not be parsed as $(c_{1,1}, c_{1,2}, c_{1,3})$ for a *first-level* ciphertext, or $(c_{1,1}, \cdots, c_{l,1}, \cdots, c_{l,5})$ for an l^{th}-*level* ciphertext $(l > 1)$, then return $\perp$. Otherwise, continue the following process:

  1. For *a first-level* ciphertext,
     (a) Verify that $e(g, c_{1,3}) = e(c_{1,1}, h_1^{H_1(c_{1,1})} h_2^{H_1(c_{1,1} \| c_{1,2})} h_3)$. If not, return $\perp$.
     (b) Otherwise, compute $m \leftarrow c_{1,2} / e(c_{1,1}, g_1^{sk_i})$.
     (c) Output m.
  2. for an l^{th}-*level* ciphertext $(l > 1)$,
     (a) Check if $e(g, c_{l,4}) = e(c_{l,1}, h_1^{H_1(c_{l,1})} h_2^{H_1(c_{l,1} \| c_{l,2} \| c_{l,3})})$. If not, return $\perp$. Otherwise,
     (b) For $\forall\ k \in [2, l]$, check $\mathcal{V}(c_{k,3}, c_{k,5}, (c_{1,1}, \cdots, c_{k,1}, c_{k,3}, c_{k,4})) = 1$.
     Whenever one of them fails, output $\perp$. Otherwise, do the following:
     (c) Compute $K_{l-1} \leftarrow c_{l,2} / e(c_{l,1}, g_1^{sk_i})$.
     (d) For i from $l-2$ down to 1, compute $K_i \leftarrow c_{i+1,2} / e(c_{i+1,1}, H_2(K_{i+1}))$.
     (e) Compute $m \leftarrow c_{1,2} / e(c_{1,1}, H_2(K_1))$.
     (f) Output m.

Consistency. Let us consider the following cases:

1. For an original (i.e., the *first-level*) ciphertext $C^{(1)} = (c_{1,1}, c_{1,2}, c_{1,3})$, we have $l = 1$. If $e(g, c_{1,3}) = e(c_{1,1}, h_1^{H_1(c_{1,1})} h_2^{H_1(c_{1,1} \| c_{1,2})} h_3)$, we have

$$m = \frac{c_{1,2}}{e(c_{1,1}, g_1^{sk_i})} = \frac{m \cdot e(g_1, pk_i^r)}{e(g^r, g_1^{x_i})} = \frac{m \cdot e(g^r, g_1^{x_i})}{e(g^r, g_1^{x_i})} = m.$$

2. For an l^{th}-*level* $(l \geq 2)$ ciphertext $C^{(l)} = (c_{1,1}, \cdots, c_{l,1}, c_{l,2}, c_{l,3}, c_{l,4}, c_{l,5})$, we have
$c_{1,1} = g^r$, $c_{1,2} = m \cdot e(g_1, pk_1)^r \cdot e(g^r, H_2(K_1) \cdot g_1^{-sk_1})$, $c_{1,3} = (h_1^{H_1(c_{1,1})} h_2^{H_1(c_{1,1} \| m \cdot e(g_1, pk_1)^r)} h_3)^r$, $\cdots$, $c_{l,1} = R_1^{(l-1)} = g^{r_{l-1}}$, $c_{l,2} = R_2^{(l-1)} = K_{l-1} \cdot e(g_1, pk_l)^{r_{l-1}}$, $c_{l,3} = R_3^{(l-1)} = svk_{P_{l-l}}$, $c_{l,4} = R_4^{(l-1)} = (h_1^{H_1(c_{l,1})} h_2^{H_1(c_{l,1} \| c_{l,2} \| c_{l,3})})^{r_{l-1}}$, $c_{l,5} = S_{l-1} = \mathcal{S}(ssk_{P_{l-1}}, (c_{1,1}, \cdots, c_{l,1}, c_{l,3}, c_{l,4}))$.
And re-encryption key $(R_1^{(l-1)}, R_2^{(l-1)}, R_3^{(l-1)}, R_4^{(l-1)}, R_5^{(l-1)})$, where $R_5^{(l-1)} = H_2(K_{l-1}) \cdot g_1^{-sk_{l-1}}$.
First we verify that
$e(g, c_{l,4}) \stackrel{?}{=} e(c_{l,1}, h_1^{H_1(c_{l,1})} h_2^{H_1(c_{l,1} \| c_{l,2} \| c_{l,3})})$.
If it holds, then for $\forall\ k \in [2, l]$, we verify that $\mathcal{V}(c_{k,3}, c_{k,5}, (c_{1,1}, \cdots, c_{k,1}, c_{k,3}, c_{k,4})) \stackrel{?}{=} 1$.
If all of these equations hold, we compute

$$K_{l-1} = \frac{c_{l,2}}{e(c_{l,1}, g_1^{sk_l})} = \frac{K_{l-1} \cdot e(g_1, pk_l)^{r_{l-1}}}{e(g^{r_{l-1}}, g_1^{sk_l})},$$

and for i from $l-2$ down to 1, compute

$$\begin{aligned} K_i &= \frac{c_{i+1,2}}{e(c_{i+1,1}, H_2(K_{i+1}))} \\ &= \frac{K_i \cdot e(g_1, pk_{i+1})^{r_i} \cdot e(g^{r_i}, R_5^{(i+1)})}{e(g^{r_i}, H_2(K_{i+1}))} \\ &= \frac{K_i \cdot e(g_1, g)^{r_i x_{i+1}} \cdot e(g^{r_i}, H_2(K_{i+1}) \cdot g_1^{-x_{i+1}})}{e(g^{r_i}, H_2(K_{i+1}))} \\ &= K_i \cdot e(g_1, g)^{r_i x_{i+1}} \cdot e(g^{r_i}, g_1^{-x_{i+1}}) = K_i \end{aligned}$$

Next, we compute m as follows:

$$\begin{aligned} m &= \frac{c_{1,2}}{e(c_{1,1}, H_2(K_1))} \\ &= \frac{m \cdot e(g_1, pk_1)^r \cdot e(g^r, H_2(K_1) \cdot (g_1^{-sk_1})}{e(g^r, H_2(K_1))} \\ &= \frac{m \cdot e(g_1, g^{x_1})^r \cdot e(g^r, H_2(K_1)) \cdot e(g^r, (g_1^{-sk_1})}{e(g^r, H_2(K_1))} \end{aligned}$$

From 1 and 2, we can draw the conclusion that our PRE scheme is consistent.

Remark. There are three elements in the original ciphertext, and $5l-2$ elements in each l^{th}-*level* ciphertext ($l > 1$) in the above scheme. The scheme permits an arbitrary number of re-encryptions on a ciphertext, with a five-element ciphertext expansion on each re-encryption. This is caused by the inclusion of a portion of the re-encryption key [8] and a signature on the newly produced ciphertext. To the best of knowledge, by now, there are no multi-use PRE schemes with no ciphertext expansion.

3. SECURITY PROOF SKETCH

THEOREM 1. *If there exists a p.p.t. adversary $\mathcal{A}$ that makes at most q_{ex} private key extraction, q_{rk} re-encryption key extraction, q_{reen} re-encryption, q_d decryption queries, and breaks our PRE scheme in the sense of IND-Pr-CCA2 with non-negligible advantage ε, then there exists a p.p.t. algorithm $\mathcal{B}$ that can solve the DBDH problem with non-negligible advantage at least $\varepsilon/e(q_{ex} + q_{rk} + q_d + q_{reen} + 1)$.*

The challenger $\mathcal{B}$ accepts as input a properly-distributed tuple $\langle \mathbb{G}_1 = \langle g \rangle, g^a, g^b, g^c, T \rangle \in \mathbb{G}_1^4 \times \mathbb{G}_T$. $\mathcal{B}$ outputs 1 if $T = e(g,g)^{abc}$, and 0 otherwise. $\mathcal{B}$ generates the scheme's system public parameters *par* as follows: (1) choose bilinear map groups $(q, g, \mathbb{G}_1, \mathbb{G}_T, e)$; (2) select $s_1, s_2, s_3 \in_R \mathbb{Z}_q^*$, and set $g_1 = g^b$, $h_1 = (g^b)^{s_1}$, $h_2 = g^{s_2}$, $h_3 = (g^b)^{-s_1 H_1(g^c)} g^{s_3}$; (3) choose a strongly unforgeable one-time signature scheme $Sig = (\mathcal{G}, \mathcal{S}, \mathcal{V})$; (4) choose two one-way, collision-resistant cryptographic hash functions H_1: $\{0,1\}^* \to \mathbb{Z}_q^*$, H_2: $\mathbb{G}_T \to \mathbb{G}_1$. Then, $par = (q, g, g_1, h_1, h_2, h_3, \mathbb{G}_1, \mathbb{G}_T, e, Sig, H_1, H_2)$ is given to the adversary $\mathcal{A}$.

- $\mathcal{O}_{pk}$: $\mathcal{B}$ selects $x_i \leftarrow_R \mathbb{Z}_q^*$ and randomly set $\alpha_i \in \{0,1\}$ such that $\Pr[\alpha_i = 1] = \gamma[2, 6]$. If $\alpha_i = 0$, $\mathcal{B}$ computes $pk_i = (g^a)^{x_i}$; Otherwise, if $\alpha_i = 1$, $\mathcal{B}$ computes $pk_i = g^{x_i}$. At last, $\mathcal{B}$ records the tuple (pk_i, x_i, α_i) in T_{pk}.
- $\mathcal{O}_{sk}$: If $\alpha_i = 0$, $\mathcal{B}$ reports failure and aborts the simulation. If $\alpha_i = 1$, $\mathcal{B}$ responds $\mathcal{A}$ with x_i.
- $\mathcal{O}_{rk}$: If $\alpha_i = 0$, $\mathcal{B}$ aborts the simulation, for he cannot compute the fourth element in the re-encryption key. Otherwise, run `RKGen`.
- $\mathcal{O}_{reen}$: If the ciphertext is valid, $\mathcal{B}$ makes an $\mathcal{O}_{rk}$ first, then corresponds $\mathcal{A}$ by executing `ReEnc`.
- $\mathcal{O}_{dec}$: If $\alpha_i = 1$, $\mathcal{B}$ decrypts the ciphertext using the secret key. Otherwise, $\mathcal{B}$ can decrypt the ciphertext using $c_{1,3}$ or $c_{l,4}$, $l > 1$.

$\mathcal{B}$ generates the challenge ciphertext as $c_{1,1}^* = g^c$, $c_{1,2}^* = m_d \cdot T^{x^*}$, $c_{1,3}^* = (g^c)^{s_2 H_1(c_{1,1}^* \| c_{1,2}^*)} (g^c)^{s_3}$.

$\mathcal{A}$ continues to make queries as above with some restrictions described by a dynamic directed graph.

4. CONCLUSION AND FUTURE WORKS

In this paper, we construct a proxy re-encryption scheme which has several traits: (1) unidirectionality; (2) multi-use; (3) collusion-safe; (4) non-interactivity; and (5) non-transitivity. Our scheme is proven to be IND-CCA2 secure under the DBDH assumption in the standard model.

Our PRE scheme is a confirmative answer to the second open problem given by Canetti and Hohenberger [4].

However, we find that (1) the ciphertext expansion is still a puzzle, and (2) the reduction is not tight. We remain these as the future works.

5. REFERENCES

[1] M. Blaze, G. Bleumer, and M. Strauss. Divertible protocols and atomic proxy cryptography. In *EUROCRYPT*, pages 127–144, 1998.

[2] D. Boneh and M. K. Franklin. Identity-based encryption from the weil pairing. In J. Kilian, editor, *CRYPTO*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer, 2001.

[3] R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In C. Cachin and J. Camenisch, editors, *EUROCRYPT*, volume 3027 of *Lecture Notes in Computer Science*, pages 207–222. Springer, 2004.

[4] R. Canetti and S. Hohenberger. Chosen-ciphertext secure proxy re-encryption. In P. Ning, S. D. C. di Vimercati, and P. F. Syverson, editors, *ACM Conference on Computer and Communications Security*, pages 185–194. ACM, 2007.

[5] T. E. Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, 1985.

[6] C. Gentry and A. Silverberg. Hierarchical id-based cryptography. In Y. Zheng, editor, *ASIACRYPT*, volume 2501 of *Lecture Notes in Computer Science*, pages 548–566. Springer, 2002.

[7] K. B. Giuseppe Ateniese and S. Hohenberger. Key-private proxy re-encryption. In *CT-RSA*, 2009.

[8] M. Green and G. Ateniese. Identity-based proxy re-encryption. In J. Katz and M. Yung, editors, *ACNS*, volume 4521 of *Lecture Notes in Computer Science*, pages 288–306. Springer, 2007.

[9] B. Libert and D. Vergnaud. Unidirectional chosen-ciphertext secure proxy re-encryption. In R. Cramer, editor, *Public Key Cryptography*, volume 4939 of *Lecture Notes in Computer Science*, pages 360–379. Springer, 2008.