Bloom フィルタを用いたマッチング数の秘匿比較

菊池 浩明† 佐久間 淳‡

† 東海大学情報通信学部通信ネットワーク工学科
106-8619 東京都港区高輪 2-3-23
kikn@tokai.ac.jp

‡ 筑波大学大学院システム情報工学研究科
305-8573 つくば市天王台 1-1-1 F934
jun@cs.tsukuaba.ac.jp

あらまし プライバシー保護データマイニングには，安全なマーケティングや匿名のヘルスケア，安全な疫学など多くの潜在的応用がある．本稿では，集合を秘匿したままで複数の集合の交わりの大きさだけを比較するプロトコルを提案する．提案方式は Bloom フィルタの内積の大きさを予測する．固定長のフィルタの為，通信効率が高い．

Privacy-Preserving Comparison of Cardinalities using Bloom Filter

Hiroaki Kikuchi† Jun Sakuma‡

†Dept. of Communication and Network Engineering,
School of Information and Telecommunication Engineering, Tokai University
2-3-23 Takanawa, Minato, Tokyo, 106-8619
‡Graduate School of SIE, Computer Science Department, University of Tsukuba
1-1-1 Tennodai, Tsukuba, 305-8573

Abstract Privacy-Preserving Data mining has many potential applications including private marketing, anonymous healthcare, and secure epidemiology. This paper proposes a new scheme for comparison of cardinalities of intersection of given pair of private subsets without revealing any element of intersections. The proposed scheme estimates the size of intersection based on the scalar product of the corresponding Bloom filters with constant size of bits. The scheme is efficient in terms of communication.

1 はじめに

リストの要素を秘匿したままでそのマッチングの数，すなわち，積集合の大きさのみを評価する 2 者間のプロトコルを考える．ここで，互いの持つ集合の如何なる要素や積集合の要素は秘匿する．この問題は，セキュアな生体認証，プライバシー保護データマイニング [2]，セキュア疫学調査 [1] などの多くの応用例に共通する基本的な要素技術の一つである．

秘匿積集合評価を安全に実行するにはいくつかの方式が知られている．準同型性を満たす公開鍵暗号による多項式評価 [5], 可換性を満たす一方向性関数を用いた秘匿積集合評価 [3] などである．しかしこれらは，集合の要素数 n に依存する通信（計算）コストがかかる．これに対して，Kantarcioglu らは，Bloom Filter を導入し，固定長の $k(< n)$ のビットへ変換した近似評価手法を提案している [10].

本研究でも，彼らと同様に Bloom Filter を効果的に用いて秘匿したままの評価を考える．Kantarcioglu らが Bloom Filter の 1 のビット数と元の積集合の大きさとの間に成立する確率の近似式に基づいて，秘匿内積評価と秘密関数計算を組み合わせたのに対し，我々の提案方式では，ベイズの定理に基づく近似式を基にする．Kantarcioglu らの方式で必要とする，秘匿積計算のプロトコルを必要としないので，計算効率がよい．また，秘匿内積プロトコルの代わりに，Bellovin らに提案された暗号化 Bloom Filter[7] をベースとしたブラインドハッシュ関数評価の技術を用いる．

安全性や要素技術の異なる 3 つの方式を提案し，その評価を与える．

2 関連研究

2.1 Bloom Filter

S を n 個の要素からなる集合 $S = \{a_1, \ldots, a_n\}$ とする．Bloom filter (BF) は，k 個の独立したハッシュ関数 $H_i : 2^* \to \{1, \ldots, m\}$, $(i = 1, \ldots, k)$ によって定められる S を表す m ビットのデータ構造である．S の BF を $B(S) = \bigcup_{a \in S} B(a)$ と定める．ここで，$B(a) = \{H_1(a), \ldots, H_k(a)\}$ とする．また，$\boldsymbol{B}$ を B から定まる m 次元ベクトル $(b_1, \ldots, b_m)$，すなわち，$i = 1, \ldots, m$ について，

$$b_i = \begin{cases} 1 & \text{if } i \in B(S), \\ 0 & \text{if } i \notin B(S), \end{cases}$$

と定める．例えば，$m = 8$ の時，$H_1(a) = 2, H_2(a) = 7$ ならば，$B(a) = \{2, 7\}$, $\boldsymbol{B}(a) = (0, 1, 0, 0, 0, 0, 1, 0)$ である．ベクトルと集合の表現は一対一に対応しており，都合のよい表記を用いる．ここで，

$$\boldsymbol{B}(S_1) \cdot \boldsymbol{B}(S_2) = |B(S_1) \cap B(S_2)|$$

であることに注意せよ．

ある要素 a が集合 S に属するか否かは，

$$\forall i = 1, \ldots, k \, H_i(a) \in B(S) \tag{1}$$

で評価する．真に $a \in S$ ならば，式 (1) は常に成立するので，偽陰性 (false negative) はないが，$a \notin S$ が式 (1) を満たす，すなわち，false positive は生じる．$B(S)$ に要素 i がない (i ビット目が 0 である) 確率は，

$$p = \left(1 - \frac{1}{m}\right)^{kn} \approx e^{-kn/m} \tag{2}$$

で与えられることが知られている [8]. 従って，偽陽性が生じる確率は，

$$p' = \left(1 - (1 - \frac{1}{m})^{kn}\right)^k \approx \left(1 - e^{-kn/m}\right)^k \tag{3}$$

で与えられる．与えられた m と n に対して，k が小さすぎると式 (1) が容易に成立してしまい，逆に大き過ぎると B がほとんど 1 で埋まってしまう．[11] によると，$k^* = \ln 2m/n$ の時最適となることが知られている．図 1 にこの確率の分布を示す．

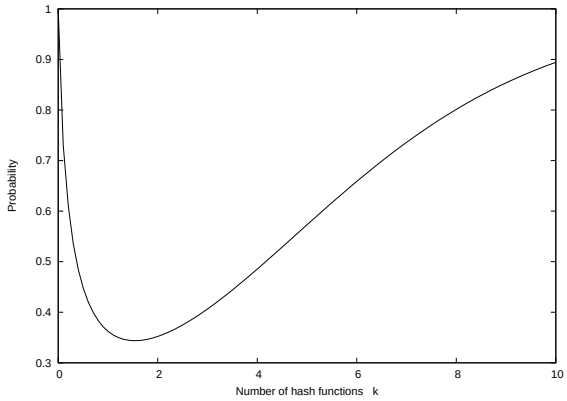

図 1: 偽陽性の確率の分布

2.2 暗号化 Bloom Filter

Bellovin と Cheswick は，Pohling-Hellman (PH) 暗号 [12] を用いて S を持つパーティ A と a を持つパーティ B が互いの値を知らせないで，$a \in S$ だけを判定するプロトコルを提案している [7]. PH 暗号は，$2q + 1 = p$ となる素数 p, q と秘密鍵 s について，平文 x の暗号文 $E(x) = x^s \bmod p$ と定める共通鍵暗号である．暗号文 c は，s の $\bmod q$ の逆元 $1/s$ を用いて，$D(c) = c^{1/s} \bmod p$ で与えられる．Bellovin らは，この可換群をなす暗号化関数をハッシュ関数

$$H(x) = x^s \bmod p$$

に利用する方式を提案している[1].

この方式を用いると，次のように値を秘匿したままでのハッシュ関数値を求めるブラインドハッシュ関数を実現し，BF を検査することが出来る．

1. A は k 個の PH 暗号の鍵 $s_1, \ldots, s_k$ を持つ．S について $B(S)$ を求め，B へ送る．
2. B は乱数 $r \bmod p - 1$ を選び，検査したい要素 x について，$y = x^r \bmod p$ を A に送る．
3. A は $y^{s_1}, \ldots, y^{s_k}$ を求めて送り返す．
4. B は，$i = 1, \ldots, k$ について，$H_i(x) = (y^{s_i})^{1/r} = x^{rs_i/r} = x^{s_i} \pmod p$ を求め，$x^{s_i} \in B(S)$ を調べる．全ての i について成立していたならば，式 (3) の偽陽性の確率で $x \in S$ である．

[1] 厳密にはここで示す方式とは異なる

2.3 積集合の大きさを比較する秘匿プロトコル

Kantarcioglu，Nix と Vaidya らは，2 者間で相関ルールをデータマイニングする目的で，Bloom Filter を使った暗号プロトコルを提案している [10]. S_A を持つ A と S_B を持つ B が，互いの集合を秘匿したままで，共通の閾値 t について，

$$X = |S_A \cap S_B| \geq t \tag{4}$$

が成立するか否かだけを求めたい．ここで，X を共通集合の大きさを取る確率変数と定義する．

BF のベクトルにおける 1 のビット数は，登録する S の大きさに比例して増える．従って，集合を直接比較する代わりに，BF に登録した結果で，

$$Y = |B(S_A) \cap B(S_B)| \geq t'$$

を評価すれば，真の積集合の大きさが予測できる．Y は，X と同様に BF の積集合の確率変数である．

Broder らによる解析結果 [8] に基づくと，式 (4) の成立は，

$$Z_A + Z_B - Z_{AB} \geq Z_A Z_B \frac{1}{m}(1 - \frac{1}{m})^{-kt}$$

と同値である．ここで，Z_A, A_B は A, B における BF の 0 のビット数 ($Z_A = m - |B(S_A)|$)，$Z_{AB} = m - |B(S_A) \cap B(S_B)| = m - Y$ である．この不等式を秘匿したままで評価する為に，秘匿内積プロトコル [13] を用いて，

$$u_1 + u_2 = \boldsymbol{B}(S_A) \cdot \boldsymbol{B}(S_B) = m - Z_{AB}$$

となる u_1, u_2 の二つの値と，

$$v_1 + v_2 = (1 - 1/m)^{-kt}/mZ_A Z_B$$

となる v_1, v_2 を秘匿積プロトコルで計算する．最後に，分散比較プロトコルを用いて，

$$(Z_A + u_1 - m) + (Z_2 + u_2) \geq (v_1 + v_2)$$

で判定する．$n = 20,000$ のデータの例で，厳密に比較をすると 27 分かかるところが，BF で近似計算をすると 4 分に削減出来ることが報告されている [10].

3 提案方式

3.1 アイデア

Kantarcioglu らの方式は巧妙で安全性が高いが，手順が複雑で比較的コストのかかる秘匿積プロトコルの実行を必要とする．そこで，$Z_{AB} = m - Y$ からベイズの定理に基づいて，$X = |S_A \cap S_B|$ を求める新たな方式 (プロトコル 1) を提案する．

更に，Bellovin らの秘匿 BF プロトコルを応用し，秘匿内積プロトコルに依存しない方式を提案する．

3.2 プロトコル 1（ベイズ応用）

n_A 個の要素を持つ A の BF において，あるビットが 0 になる確率は，式 (2) によって，$q_A = (1 - 1/m)^{kn_A}$. 同様に，B において，$q_B = (1-1/m)^{kn_B}$. この時，$A \cap B$ の BF においてビットが 1 になる確率は，$q = (1 - q_A)(1 - q_B)$. この時，$Y = |B(S_A) \cap B(S_B)|$ が生じる確率は，q による 2 項分布で与えられる．すなわち，

$$Pr[Y = y|X = x] = \begin{pmatrix} m - x \\ y - x \end{pmatrix} q^{y-x}(1 - q)^{m-y}.$$

$n_A = 10$, $n_B = 8$, $k = 3$, $m = 4$ の時，$q_A = 0.47$, $q_B = 0.54$, $q = 0.24 \approx 1/2^2$ であり，$X = 4$ の時の分布を図 2 に示す．真の積集合の大きさ X に対して，BF の積集合の大きさ Y は 13 をピークとした分布を示している．

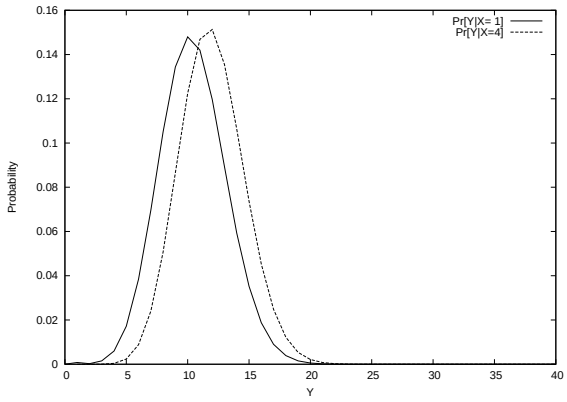

図 2: 確率分布 $Pr[Y|X = 1]$ と $Pr[Y|X = 4]$

2 項分布の平均 $E(X) = mq$ であることを利用すると，Y の期待値は

$$E[Y|X = x] = x + (m - x)q = f(x)$$

であり，これを $\hat{Y} = f(x)$ と置く．逆に，BF の積集合の大きさ Y が与えられたときの真の X は，f の逆関数で，

$$\hat{X} = f^{-1}(y) = \frac{y - mq}{1 - q} \tag{5}$$

で与えられる．例えば，先の数値例では，$\hat{X} = 4$ で近似出来る．

より厳密に，ベイズの定理に基づいて推定すると，

$$\begin{aligned} Pr[X|Y=y] &= \frac{Pr[Y=y|X]Pr[X]}{Pr[Y=y]} \\ &= \frac{Pr[Y=y|X]Pr[X]}{\sum_X Pr[Y=y|X=x]Pr[X=x]} \\ &= \frac{Pr[Y=y|X]1/n}{\sum_X Pr[Y=y|X=x]1/n} \end{aligned}$$

で与えられる．ただし，$n = \min(n_A, n_B)$ であり，両パーティ間で同意を取って共有しておくことが必要である．数値例から算出した確率分布を図 3 に示す．

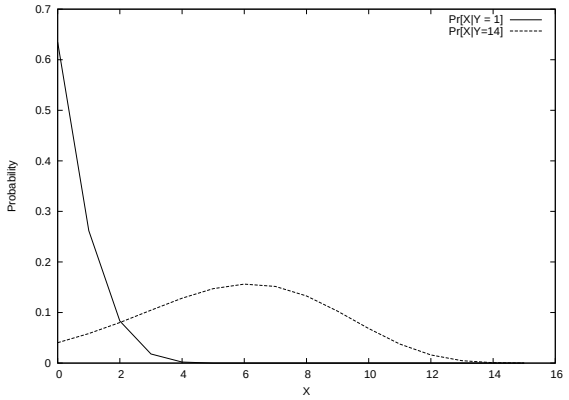

図 3: ベイズ推定した X の確率分布 $Pr[X|Y]$

最後に，確率分布から期待値 $E[X|Y] = \sum_x xPr[X = x|Y]$ を求めるか，最尤値 $L[X|Y] = \mathrm{argmax}_x Pr[X = x|Y]$ で真の積集合の大きさ X を推定する．図 4 に，$n = 10$ の時の BF の積集合の大きさ Y から推定される真の積集合の大きさ X を示す．最尤値は，式 (5) で算出される $\hat{X}$ の振る舞いとほぼ同じであり，簡単に推定できるが，期待値よりも Y に対する変化が著しい．

(プロトコル 1)

1. A と B は，$B(S_A), B(S_B)$ をそれぞれ計算する．
2. セキュア内積プロトコルを適用し，$u_A + u_B = \boldsymbol{B}(S_A) \cdot \boldsymbol{B}(S_B)$ となる u_A と u_B を求める．
3. 判定するしきい値 θ について，$Pr[Y|X = \theta]$ を求める．$X \geq \theta$ と同値な $Y \geq \theta'$ を求め，秘匿比較プロトコルを適用し，$Y \geq \theta'$ を判定する．

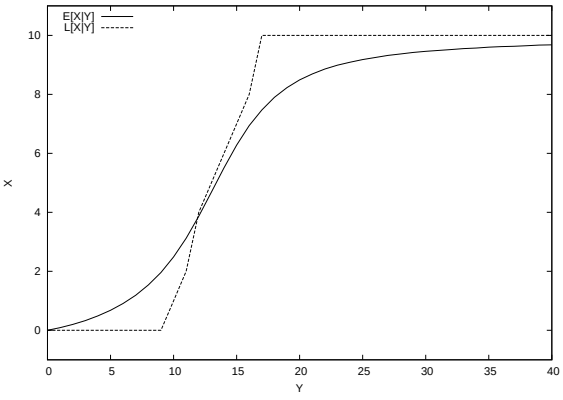

図 4: 積集合の大きさ X の推定．期待値 $E[X|Y]$ と最尤値 $L[X|Y]$

3.3 プロトコル 2 (秘匿 BF)

ハッシュ関数が分からなければ，$B(S)$ から S を推定することは困難であることを仮定し，Bellovin らの暗号化 BF プロトコルを応用して，B が代表して比較を行うプロトコルを構成する．

(プロトコル 2)

1. A は k 個の PH 暗号の鍵 $s_1, \ldots, s_k$ を持つ．S_A について $B(S_A)$ を求め，B へ送る．
2. B は乱数 $r \bmod p-1$ を選び，全ての $b_j \in S_B$ に　ついて，$y_j = b_j^r \bmod p$ を求め A に送る．
3. A は $y_j^{s_1}, \ldots, y_j^{s_k}$ を求めて送り返す．
4. B は，$i = 1, \ldots, k$ について，$H_i(b_j) = (y_j^{s_i})^{1/r} = x^{rs_i/r} = b_j^{s_i} \pmod p$ を求め，$Y = |\{j|b_j^{s_i} \in B(S_A)\}|$ を求める．プロトコル 1 と同様に，$E[X|Y]$ または $L[X|Y]$ で $\hat{X}$ を推定し，$\hat{X} > \theta$ を判定する．

3.4 プロトコル 3 (分散型秘匿 BF)

プロトコル 2 では，B だけが判定結果を知ってしまう．しかも，ステップ 1 で得られた $Y = |B(S_A)|$ から，X の推測もできるので，A の集合の大きさ n_A も分かってしまう．そこで，これを安全に行うため，A, B に分散して計算する方式に拡張する．BF の計算が，集合和に対し準同型性を満たすことを利用して，分散したままでプロトコル 2 を適用する．

(プロトコル 3)

1. A と B が同意のもと，ハッシュ関数 $h : 2^* \rightarrow \{1, 2\}$ をを決めて，S_A を $S_{A1} = \{a \in S_A | h(a) = 1\}$, $S_{A2} = S_A - S_{A1}$ の二つに分割する．S_B も同様にする．

2. S_{A1} と S_{B1} についてプロトコル 2 を用いて，B が $Y_1 = |B(S_{A1} \cap B(S_{B1})|$ を求め，$\hat{x_1}$ を得る．

3. S_{A2} と S_{B2} についてプロトコル 2(対称にして) を用いて，A が $Y_2 = |B(S_{A2} \cap B(S_{B2})|$ を求め，$\hat{x_2}$ を得る．

4. A, B は秘匿比較プロトコルを用いて，$\hat{x_1} + \hat{x_2} \geq \theta$ を判定する．

4 評価

4.1 安全性

プロトコル 1 の安全性は，秘匿内積プロトコルの安全性に基づく．プロトコル 2 を証明するには，PH 暗号の秘匿性のもとで，比較する集合の識別ができないかを示さなくてはならない．

プロトコル 3 は，プロトコル 2 に対して，比較の結果をそれぞれが分からないことを保証しなくてはならない．計算は分散して行っているが，ハッシュ関数 h が一様ならば，A, B の計算結果も近く，$\hat{x_1} \approx \hat{x_2}$ となってしまう課題がある．

4.2 近似精度

BF そのものが持つ誤差は，k を増やすことで小さくすることができる．提案方式では，$n_A \approx n_B$ を仮定しているが，この差が大きい時に生じる誤差を見積もる必要があるだろう．

4.3 計算，通信コスト

S_A, S_B を秘匿して積集合を計算するプロトコルには，Freedman らの多項式評価に基づくもの [5] や，Agrawal らの可換な一方向性関数に基づく [3] がよく知られている．いずれも，要素数 n に依存するプロトコルだが，BF ではそれらを，m のオーダーにする．従って，いずれも高い通信効率が期待できる．ただし，m に比例する計算コストのプロトコル 1 に対して，プロトコル 2 と 3 は k 種類の値 (BF の 1 の要素，$k < m$) だけを交換すればよいが，ブラインドハッシュ関数評価の為に n に依存する項が存在する．

その反面，n 個の要素を BF に登録する際に n の計算量がかかる．プロトコル 1 は汎用のハッシュ関数を使えばいいので，そのコストは無視できる．プロトコル 2 と 3 は，ハッシュ関数の評価に PH 暗号が必要なので比較的重いコストを考慮しなくてはならない．

以上の性能を表 1 に整理する．従来方式の代表として，FNP[5] と KNV[10] のコストと比較も示す．ここで，c は計算コストであり，c_e, c_d, c_p, c_h はそれぞれ，暗号化，復号，べき乗，ハッシュ関数（通常のもの) の計算コストである．$c_e \doteq c_d \doteq c_p \gg c_h$ である．計算コストでは，ℓ_p は暗号文のサイズであり，$\ell_p = 2048$ bit と考えてよい．(PH 暗号の時のサイズはより小さくてもよい)．KNV[10] の性能はプロトコル 1 とほぼ同様であるが，秘匿積評価のための処理の有無に差がある．

5 結論

3 種類の秘匿積集合プロトコルを提案した．提案方式の安全性やパフォーマンスのより厳密な評価を今後の課題とする．

参考文献

[1] 統計局政策統括官・(統計基準担当) 統計研修所, 厚生労働省大臣官房統計情報部人口動態・保健統計課「人口動態統計」

[2] 菊池浩明, 香川大介, 石井一彦, 寺田雅之, 本郷節之, “組織間プライバシー保護データマイニングの考察”, SCIS2010.

[3] Rakesh Agrawal, Alexandre Evfimievski, and Ramakrishnan Srikant, “Information shareing across private databases”, in proc. of ACM SIGMOD Intl. Conf. on Management of Data, 2003.

[4] Vaidya, J. and C. Clifton, “Privacy preserving association rule mining in vertically partitioned data”, The Eighth ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data

表 1: 提案方式の性能評価

	FNP[5]	Protocol 1	Protocol 2	Protocol 3
原理	秘匿多項式	秘匿内積, SFE	ブラインドハッシュ	ブラインドハッシュ, SFE
計算 A	$c_e n_A + c_d n_A n_B$	$c_h n_A + c_e m + c_d$	$c_p(k n_A + k n_B)$	$c_p(k+1/2)n_A + k n_B/2$
コスト B	$c_p n_A n_B$	$c_h n_B + c_p m + c_e$	$c_p n_B(1+k)$	$c_p(k+1/2)n_B + k n_A/2$
通信コスト	$\ell_p(n_A n_B)$	$\ell_p m + 1$	$k + \ell_p(k+1)n_B$	$2k + \ell_p(k+1)(n_A+n_B)/2$

Mining, SIGKDD, ACM Press, Edmonton, Canada, pp. 639-644, 2002.

[5] M. J. Freedman, K. Nissim, and B. Pinkas, “Efficient private matching and set intersection”, EUROCRYPT 2004, LNCS 3027, pp. 1?19, Springer-Verlag, 2004.

[6] Dahlia Malkhi, Noam Nisan, Benny Pinkas, and Yaron Sella, “Fairplay - A Secure Two-Party Computation System”, Usenix Security Symposium, 2004.

[7] S.M. Bellovin, W.R. Cheswick, “Privacy-Enhanced Searches Using Encrypted Bloom Filters”, Cryptology ePrint Archive, 2004/022.

[8] A. Broder, M. Mitzenmacher, “Network Applications of Bloom Filters: A Survey”, Internet Math, Volume 1, Number 4 (2003), 485-509.

[9] Ryo Nojima and Youki “Cryptographically Secure Bloom-Filters”, Trans. Data Privacy, Vol. 2, No. 2 , pp. 131-139. 2009.

[10] Murat Kantarcioglu, Robert Nix and Jaideep Vaidya, “An Efficient Approximate Protocol for Privacy-Preserving Association Rule Mining”, 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD 2009), LNCS 5476, Springer, pp. 515-524, 2009.

[11] Li Fan, Pei Cao, Jussara Almeida, and Andrei Z. Broder, “Summary cache: a scalable wide-area web cache sharing protocol”, IEEE/ACM Trans. Netw. Vol. 8, No. 3, pp. 281-293, 2000.

[12] S. Pohlig and M. Hellman, “An Improved Algorithm for Computing Logarithms over $GF(p)$ and its Cryptographic Significance”, IEEE Transactions on Information Theory (24), pp. 106-110, 1078.

[13] Bart Goethals, Sven Laur, Helger Lipmaa and Taneli Mielikainen, “On Private Scalar Product Computation for Privacy-Preserving Data Mining”, The 7th Annual International Conference in Information Security and Cryptology (ICISC 2004), Vol. 3506 of LNCS, pp. 104-120, 2004.