A protocol for K-multiple substring matching *

Vadym Fedyukovych Vitaliy Sharapov

August 18, 2008

Abstract

A protocol is introduced to show K copies of a pattern string are embedded is a host string. Commitments to both strings, and to offsets of copies of the pattern in the host is input of Verifier. Protocol does not leak useful information about strings, and is zero knowledge.

1 Preliminaries

We consider a string as a set of tuples of integers for each character and it's position. This can be elaborated by introducing a set of allowed characters, and require positions to be continuous.

Definition 1 (Polynomial representation)**.** *String characteristic polynomial* is a mapping from all sets of character-position tuples to a ring of polynomials over integers:

$$S : \{(c_j, i_j)\} \to F(x, y; S) \tag{1}$$

$$F(x, y; S) = \prod_{j=1}^{|S|}(1 + xc_j + yi_j) \tag{2}$$

A related definition for graph characteristic polynomial appeared with protocols for graph isomorphism and vertex coloring [Fed08].

Lemma 1 (Schwartz-Zippel [Sch80], a case of a univariate polynomial)**.** *Probability to choose a root of a non-zero polynomial $f(z)$ of degree at most d by choosing some $z = c$ at random from a set D*

$$\Pr[f(c) = 0 \mid c \in_R D] \leq \frac{d}{|D|}$$

*Extended abstract. This report to appear in ITaS'08 (in Russian).

A *commitment scheme* is a tuple of algorithms $Gen(), Commit(), Open()$ such that *binding* and *hiding* holds. An integer commitment scheme [DF02] suggest a group of a hidden order. We use this scheme with an additional requirement of avoiding small divisors of order of the group by using strong primes to produce module N. Protocols for this scheme achieve soundness on assumption of hardness of Strong RSA problem, as well as statistically indistinguishable simulated transcripts.

2 Protocol design

Let $\{(c_{Pj}, i_{Pj})\}$ be characters of pattern and their positions, $\{(c_{Hj}, i_{Hj})\}$ be characters and positions of host, $\{o_k\}$ be offsets of copies of pattern at the host, $\{r_j\}$ be flags assigned to characters of host. Let $\{C_{Pj}\}$, $\{I_{Pj}\}$, $\{C_{Hj}\}$, $\{I_{Hj}\}$, $\{O_k\}$, $\{R_j\}$ be responses of a variant of Schnorr protocol [Oka92] with challenges (e, d, s) and initial random coins $\{\alpha_{Pj}\}$, $\{\beta_{Pj}\}$, $\{\alpha_{Hj}\}$, $\{\beta_{Hj}\}$, $\{\gamma_k\}$, $\{\rho_j\}$:

$$C_{Pj} = ec_{Pj} + \alpha_{Pj} \qquad I_{Pj} = di_{Pj} + \beta_{Pj} \tag{3}$$

$$C_{Hj} = ec_{Hj} + \alpha_{Hj} \qquad I_{Hj} = di_{Hj} + \beta_{Hj} \tag{4}$$

$$O_k = do_k + \gamma_k \tag{5}$$

$$R_j = sr_j + \rho_j \tag{6}$$

Definition 2. *Pattern string verification polynomial* is a mapping **from** all sets of tuples of integers $\{(C_{Pj}, I_{Pj})\}$ that are protocol responses for characters of pattern and their positions, and all sets of integers $\{O_k\}$ that are responses for pattern positions in the host, and for protocol challenges e, d **to** ring of polynomials over integers:

$$F_P(x, y; S_P) = \prod_{k=1}^{K} \prod_{j=1}^{L_P} (ed + xdC_{Pj} + ye(I_{Pj} + O_k)) \tag{7}$$

Definition 3. *Host string verification polynomial* is a mapping **from** all sets of tuples of integers $\{(C_{Hj}, I_{Hj})\}$ that are protocol responses for characters of host and their positions, and all sets of integers $\{R_j\}$ that are responses for flags, and for protocol challenges s, e, d **to** ring of polynomials over integers:

$$F_H(x, y; S_H) = \prod_{j=1}^{L_H} (eds + R_j(xdC_{Hj} + yeI_{Hj})) \tag{8}$$

We assign flags $r_j = 1$ to characters of K non-overlapping copies of pattern string in the host string, and $r_j = 0$ to all other characters of host. We say

$$S_F = \{j \mid r_j = 1\}$$

Consider expansion coefficients as follows:

$$F_H(x,y;S_H) = s^{L_H} F'_H(x,y) + \sum_{t=0}^{L_H-1} s^t v_t \tag{9}$$

$$F'_H(x,y) = (ed)^{L_H - KL_P} \prod_{j \in S_F} (ed + xdC_{Hj} + yeI_{Hj}) \tag{10}$$

$$F'_H(x,y) = (ed)^{L_H - n} \sum_{l=0}^{n} e^l a_l(x,y) \qquad \text{where } n = KL_P \tag{11}$$

$$a_n(x,y) = \sum_{m=0}^{n} d^m b_m(x,y) \tag{12}$$

$$b_n(x,y) = \prod_{k=1}^{K} \prod_{j=1}^{L_P} (1 + xc_{Pj} + y(i_{Pj} + o_k)) \tag{13}$$

Product in (10) is over characters of K copies of pattern in the host. Position of a character in the host was replaced at (13) with position of a matching character in the pattern and offset of a copy of the pattern.

Consider another set of expansion coefficients:

$$F_P(x,y;S) = \sum_{l=0}^{n} e^l a'_l(x,y) \tag{14}$$

$$a'_n(x,y) = \sum_{m=0}^{n} d^m b'_m(x,y) \tag{15}$$

$$b'_n(x,y) = \prod_{k=1}^{K} \prod_{j=1}^{L_P} (1 + xc_{Pj} + y(i_{Pj} + o_k)) \tag{16}$$

It follows that

Lemma 2. *There are at least K non-overlapping copies of the pattern string in the host string if, and only if*

$$b_n(x,y) \equiv b'_n(x,y) \tag{17}$$

holds for top expansion coefficients of pattern and host verification polynomials, and for some assignments of flags $r_j \in \{0,1\}$ to characters of the host.

Backward statement can be shown due to unique decomposition of verification polynomials into relatively prime linear polynomials.

Verifier tests that (17) holds by choosing $x = x_c, y = y_c$ at random, and testing

$$b_n(x_c,y_c) = b'_n(x_c,y_c) \tag{18}$$

with our protocol.

3 Protocol

Common input is commitment scheme parameters: (N, g, h), commitment to pattern string:

$$\{(W_{Pj}, M_{Pj})\}, \quad j = 1 \ldots L_P$$

to host string:

$$\{(W_{Hj}, M_{Hj})\}, \quad j = 1 \ldots L_H$$

and to pattern offsets in host string:

$$\{Q_k\}, \quad k = 1 \ldots K$$

Auxiliary input of Prover is characters and their positions of both strings, offsets of pattern copies in the host, as well as random coins used to produce commitments:

$$\begin{gathered} \{(c_{Pj}, i_{Pj})\}, \quad \{(c_{Hj}, i_{Hj})\}, \quad \{o_k\} \\ \{(\theta_{Pj}, \phi_{Pj})\}, \quad \{(\theta_{Hj}, \phi_{Hj})\}, \quad \{\lambda_k\} \end{gathered}$$

such that

$$\begin{gathered} W_{Pj} = g^{c_{Pj}} h^{\theta_{Pj}}, \quad M_{Pj} = g^{i_{Pj}} h^{\phi_{Pj}} \\ W_{Hj} = g^{c_{Hj}} h^{\theta_{Hj}}, \quad M_{Hj} = g^{i_{Hj}} h^{\phi_{Hj}} \\ Q_k = g^{o_k} h^{\lambda_k} \end{gathered}$$

Prover shows there are at least K copies of pattern string in the host string as follows:

1. Prover assigns flags $r_j = 1$ to characters of text string that are copies of pattern string, and $r_j = 0$ to all other characters of text string. Prover chooses random coins $\{\omega_j\}$, produces commitments:

   $$L_j = g^{r_j} h^{\omega_j} \qquad j = 1 \ldots L_H \tag{19}$$

   Prover sends $\{L_j\}$ to Verifier.

2. Verifier chooses a challenge (x_c, y_c) from the interval at random, and sends it to Prover.

3. Prover chooses random coins

   $$\{(\beta_{Hj}, \mu_{Hj})\}, \quad \{(\beta_{Pj}, \mu_{Pj})\}, \quad \{(\gamma_k, \chi_k)\} \tag{20}$$

produces expansion coefficients $\{b_m\}$, $\{b'_m\}$:

$$\prod_{j=1}^{n}(z + x_c z c_{Hj} + y_c(z i_{Hj} + \beta_{Hj})) = \sum_{m=0}^{n} z^m b_m \tag{21}$$

$$\prod_{k=1}^{K}\prod_{j=1}^{L_P}(z + x_c z c_{Pj} + y_c((z i_{Pj} + \beta_{Pj}) + (z o_k + \gamma_k))) = \sum_{m=0}^{n} z^m b'_m \tag{22}$$

chooses random coins $(\zeta_m, \zeta'_m, \psi, \tau, \tau')$, produces commitments:

$$B_m = g^{b_m} h^{\zeta_m}, \quad B'_m = g^{b'_m} h^{\zeta'_m} \quad m = 0 \ldots n \tag{23}$$

$$Y_{Pj} = g^{\beta_{Pj}} h^{\mu_{Pj}} \qquad j = 1 \ldots L_P \tag{24}$$

$$Y_{Hj} = g^{\beta_{Hj}} h^{\mu_{Pj}} \qquad j = 1 \ldots L_H \tag{25}$$

$$X_k = g^{\gamma_k} h^{\chi_k} \qquad k = 1 \ldots K \tag{26}$$

$$Z = g^{\psi} h^{\tau} \tag{27}$$

$$Z' = g^{\psi} h^{\tau'} \tag{28}$$

Prover sends $\{B_m\}$, $\{B'_m\}$, $\{Y_{Pj}\}$, $\{Y_{Hj}\}$, $\{X_k\}$, Z, Z' to Verifier.

4. Verifier chooses a non-zero challenge d at random from the interval, and sends it to Prover.

5. Prover produces responses:

$$I_{Hj} = d i_{Hj} + \beta_{Hj}, \quad \Phi_{Hj} = d\phi_{Hj} + \mu_{Hj}, \quad j = 1 \ldots L_H \tag{29}$$

$$I_{Pj} = d i_{Pj} + \beta_{Pj}, \quad \Phi_{Pj} = d\phi_{Pj} + \mu_{Pj}, \quad j = 1 \ldots L_P \tag{30}$$

$$O_k = d o_k + \gamma_k, \quad \Lambda_k = d\lambda_k + \chi_k, \quad k = 1 \ldots K \tag{31}$$

$$\Psi = d b_n + \psi, \quad \Gamma = d\zeta_n + \tau, \quad \Gamma' = d\zeta'_n + \tau' \tag{32}$$

chooses random coins $(\{(\alpha_{Hj}, \eta_{Hj})\}, \{(\alpha_{Pj}, \eta_{Pj})\})$, produces expansion coefficients $\{a_n\}$, $\{a'_n\}$:

$$\prod_{j=1}^{KL_P}(zd + x_c d(z c_{Hj} + \alpha_{Hj}) + y_c z(I_{Hj} + d o_k)) = \sum_{l=0}^{n} z^l a_l \tag{33}$$

$$\prod_{k=1}^{K}\prod_{j=1}^{L_P}(zd + x_c d(z c_{Pj} + \alpha_{Pj}) + y_c z(I_{Pj} + O_k)) = \sum_{l=0}^{n} z^l a'_l \tag{34}$$

chooses random coins $\{\pi_l\}, \{\pi'_l\}$, produces commitments:

$$A_l = g^{a_l} h^{\pi_l}, \quad A'_l = g^{a'_l} h^{\pi'_l} \quad l = 0 \ldots (n-1) \tag{35}$$

$$T_{Pj} = g^{\alpha_{Pj}} h^{\eta_{Pj}} \qquad j = 1 \ldots L_P \tag{36}$$

$$T_{Hj} = g^{\alpha_{Hj}} h^{\eta_{Hj}} \qquad j = 1 \ldots L_H \tag{37}$$

Prover sends $\{I_{Hj}\}$, $\{\Phi_{Hj}\}$, $\{I_{Pj}\}$, $\{\Phi_{Pj}\}$, $\{O_k\}$, $\{\Lambda_k\}$, Ψ, $\{A_l\}$, $\{A'_l\}$, $\{T_{Pj}\}$, $\{T_{Hj}\}$ to Verifier.

6. Verifier chooses a non-zero challenge e at random from the interval, and sends it to Prover.

7. Prover produces responses:

$$C_{Hj} = ec_j + \alpha_{Hj}, \quad \Theta_{Hj} = e\theta_{Hj} + \eta_{Hj} \tag{38}$$

$$C_{Pj} = ec_j + \alpha_{Pj}, \quad \Theta_{Pj} = e\theta_{Pj} + \eta_{Pj} \tag{39}$$

chooses random coins $\{\rho_j\}$, produces expansion coefficients $\{v_t\}$:

$$\prod_{j=1}^{L_H}(edz + (zr_j + \rho_j)(x_c dC_{Hj} + y_c eI_{Hj})) = \sum_{t=0}^{L_H} z^t v_t \tag{40}$$

produces expansion coefficients $\{(u_{1j}, u_{0j})\}$:

$$(zr_j + \rho_j)(z(r_j - 1) + \rho_j) = u_{1j}z + u_{0j} \tag{41}$$

chooses random coins $\{\sigma_t\}, \{\delta_j\}, \{(\nu_{1j}, \nu_{0j})\}$, produces commitments:

$$V_t = g^{v_t}h^{\sigma_t} \qquad t = 0 \ldots (L_H - 1) \tag{42}$$

$$N_j = g^{\rho_j}h^{\delta_j} \qquad j = 1 \ldots L_H \tag{43}$$

$$E_{1j} = g^{u_{1j}}h^{\nu_{1j}}, \quad E_{0j} = g^{u_{0j}}h^{\nu_{0j}} \tag{44}$$

Prover sends $\{C_{Hj}\}, \{\Theta_{Hj}\}, \{C_{Pj}\}, \{\Theta_{Pj}\}, \{V_t\}, \{N_j\}, \{(E_{1j}, E_{0j})\}$ to Verifier.

8. Verifier chooses a non-zero challenge s at random from the interval, and sends it to Prover.

9. Prover produces responses

$$R_j = sr_j + \rho_j, \quad j = 1 \ldots L_H \tag{45}$$

$$\Omega_j = s\omega_j + \delta_j \tag{46}$$

$$\Xi_j = s\nu_{1j} + \nu_{0j} \tag{47}$$

$$\Delta_P = e^n \sum_{m=0}^{n} d^m \zeta'_m + \sum_{l=0}^{n-1} e^l \pi'_l \tag{48}$$

$$\Delta_H = s^{L_H}(e^n \sum_{m=0}^{n} d^m \zeta_m + \sum_{l=0}^{n-1} e^l \pi_l) + \sum_{t=0}^{L_H-1} s^t \sigma_t \tag{49}$$

Prover sends $\{R_j\}, \{\Omega_j\}, \{\Xi_j\}, \Delta_P, \Delta_H$ to Verifier.

10. Verifier tests responses to fit commitments:

$$g^{C_{Pj}} h^{\Theta_{Pj}} W_{Pj}^{-e} = T_{Pj} \qquad g^{I_{Pj}} h^{\Phi_{Pj}} M_{Pj}^{-d} = Y_{Pj} \tag{50}$$

$$g^{C_{Hj}} h^{\Theta_{Hj}} W_{Hj}^{-e} = T_{Hj} \qquad g^{I_{Hj}} h^{\Phi_{Hj}} M_{Hj}^{-d} = Y_{Hj} \tag{51}$$

$$g^{O_k} h^{\Lambda_k} Q_k^{-d} = X_k \tag{52}$$

$$g^{R_j} h^{\Omega_j} L_j^{-s} = N_j \tag{53}$$

tests flags to be from the right set ($r_j \in \{0,1\}$):

$$g^{-R_j(R_j-s)} h^{-\Xi_j} E_{1j}^{s} E_{0j} = 1 \tag{54}$$

produces

$$U_P = \prod_{k=1}^{K} \prod_{j=1}^{L_P} (ed + x_c d C_j + y_c e (I_j + O_k)) \tag{55}$$

$$U_H = (ed)^{-(L_H - n)} \prod_{j=1}^{L_H} (eds + R_j (x_c d C_j + y_c e I_j)) \tag{56}$$

Verifier accepts if

(a) $\{A'_l\}, \{B'_m\}$ are commitments to expansion coefficients of pattern verification polynomial:

$$g^{-U_P} h^{-\Delta_P} \left(\prod_{m=0}^{n} (B'_m)^{d^m} \right)^{e^n} \prod_{l=0}^{n-1} (A'_l)^{e^l} = 1 \tag{57}$$

(b) $\{A_l\}, \{B_m\}$ are commitments to expansion coefficients of host verification polynomial:

$$g^{-U_H} h^{-\Delta_H} \left(\left(\prod_{m=0}^{n} (B_m)^{d^m} \right)^{e^n} \prod_{l=0}^{n-1} (A_l)^{e^l} \right)^{s^{L_H}} \prod_{t=0}^{L_H - 1} (V_t)^{s^t} = 1 \tag{58}$$

(c) top expansion coefficients committed at B_n, B'_n are the same ($b_n = b'_n$):

$$g^{\Psi} h^{\Gamma} B_n^{-d} = Z \tag{59}$$

$$g^{\Psi} h^{\Gamma'} (B'_n)^{-d} = Z' \tag{60}$$

4 Protocol properties

It is clear honest Verifier always accepts for an honest Prover and commitments such that there are at least K copies of pattern in the host.

Lemma 3 (Soundness)**.** *Probability for an honest Verifier to accept for any polynomial Prover and any commitments such that there are no or less than K copies of pattern string in host string while running protocol shown in section 3 is at most* $\frac{2KL_P+L_H}{q}$.

Lemma 4 (Zero knowledge)**.** *Protocol shown in section 3 has a simulator, and is honest verifier statistical zero knowledge.*

Proof. Consider a candidate simulator algorithm as follows. Given commitment scheme parameters (N, g, h), challenges (x_c, y_c, d, e, s), and commitments $\{(W_{Pj}, M_{Pj})\}$, $\{(W_{Hj}, M_{Hj})\}$, $\{Q_k\}$, Verifier:

1. chooses group elements: $\{E_{1j}\}$, $\{B'_m\}$, $\{A'_l\}_{l=1\ldots n-1}$, $\{B_m\}$, $\{A_l\}$, $\{V_t\}_{t=1\ldots L_H-1}$ at random;
2. chooses some $\{C_{Pj}\}$, $\{\Theta_{Pj}\}$, $\{I_{Pj}\}$, $\{\Phi_{Pj}\}$, $\{C_{Hj}\}$, $\{\Theta_{Hj}\}$, $\{I_{Hj}\}$, $\{\Phi_{Hj}\}$, $\{O_k\}$, $\{\Lambda_k\}$, $\{R_j\}$, $\{\Omega_j\}$, Δ_P, Δ_H, Ψ, Γ, Γ' at random;
3. produces

$$T_{Pj} = g^{C_{Pj}} h^{\Theta_{Pj}} W_{Pj}^{-e} \qquad Y_{Pj} = g^{I_{Pj}} h^{\Phi_{Pj}} M_{Pj}^{-d} \tag{61}$$

$$T_{Hj} = g^{C_{Hj}} h^{\Theta_{Hj}} W_{Hj}^{-e} \qquad Y_{Hj} = g^{I_{Hj}} h^{\Phi_{Hj}} M_{Hj}^{-d} \tag{62}$$

$$X_k = g^{O_k} h^{\Lambda_k} Q_k^{-d} \tag{63}$$

$$N_j = g^{R_j} h^{\Omega_j} L_j^{-s} \tag{64}$$

$$E_{0j} = g^{R_j(R_j-s)} h^{\Xi_j} E_{1j}^{-s} \tag{65}$$

$$U_P = \prod_{k=1}^{K} \prod_{j=1}^{L_P} (ed + x_c d C_j + y_c e(I_j + O_k)) \tag{66}$$

$$U_H = (ed)^{-(L_H-n)} \prod_{j=1}^{L_H} (eds + R_j(x_c d C_j + y_c e I_j)) \tag{67}$$

$$A'_0 = g^{U_P} h^{\Delta_P} \left(\prod_{m=0}^{n} (B'_m)^{d^m} \right)^{-e^n} \prod_{l=1}^{n-1} (A'_l)^{-e^l} \tag{68}$$

$$V_0 = g^{U_H} h^{\Delta_H} \left(\left(\prod_{m=0}^{n} (B_m)^{d^m} \right)^{e^n} \prod_{l=0}^{n-1} (A_l)^{e^l} \right)^{-s^{L_H}} \prod_{t=1}^{L_H-1} (V_t)^{-s^t} \tag{69}$$

$$Z = g^{\Psi} h^{\Gamma} B_n^{-d} \qquad Z' = g^{\Psi} h^{\Gamma'} (B'_n)^{-d} \tag{70}$$

Transcript simulated this way is statistically indistinguishable from all protocol transcripts with the same challenges. □

5 Discussion

Protocol introduced can be extended to show matching for 2D and 3D objects. Approximate matching can be another extension. Protocols of this type may be developed for similar statements regarding pattern matching. Protocol is expected to be useful in bioinformatics.

References

[DF02] Ivan Damgård and Eiichiro Fujisaki. A statistically-hiding integer commitment scheme based on groups with hidden order. In *ASIACRYPT*, pages 125–142, 2002.

[Fed08] Vadym Fedyukovych. Protocols for graph languages. In *(submitted)*, 2008.

[Oka92] Tatsuaki Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In *CRYPTO*, pages 31–53, 1992.

[Sch80] J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, 1980.