

# Argument of knowledge of a bounded error

Vadym Fedyukovych

August 18, 2008

## Abstract

A protocol is introduced to show knowledge of a codeword of Goppa code and Goppa polynomial. Protocol does not disclose any useful information about the codeword and polynomial coefficients. A related protocol is introduced to show Hamming weight of an error is below a threshold. Protocol does not disclose codeword and weight of the error. Verifier only uses commitments to codeword components and coefficients while testing validity of statements. Both protocols are honest verifier zero knowledge.

## 1 Introduction

Approximate matching over hidden data is an important practical problem. In particular, it may be challenging to show validity of a statement about some data while keeping the data private. We are interested in producing verifiable statements about private data. The well-known technique to produce verifiable statements is electronic signatures. However, most signatures require exact matching for data signed and signing keys.

We introduce protocols to show that values committed are a codeword of Goppa code and coefficients of Goppa polynomial, and that error weight in a damaged codeword is below a threshold.

## 2 Preliminaries

Let  $q$  be a large prime,  $\mathbb{F}_q$  be a finite field,  $\mathbb{F}_q[z]$  be polynomial ring over  $\mathbb{F}_q$ . Let  $g(z) \in \mathbb{F}_q[z]$  be a non-zero polynomial of degree  $T$ .

A *codeword of Goppa code* [8] is a  $N$ -tuple  $B = \{b_1 \dots b_N\}$ ,  $b_j \in \mathbb{F}_q$  such that

$$\sum_{j=1}^N \frac{b_j}{z - a_j} \equiv 0 \pmod{g(z)} \quad (1)$$

for a set  $A = \{a_1, \dots, a_N\}$  of different  $a_j \in \mathbb{F}_q$ . Polynomial  $g(z)$  is *Goppa polynomial*, and  $A$  is a *support set*.

Let  $h, f \in \mathbb{G}$  be two elements of order  $q$  such that  $\log_f(h)$  is not known to Prover. Pedersen commitment [9] to a value  $x \in \mathbb{F}_q$  with an accessory value  $r \in \mathbb{F}_q$  is (multiplicative notation)

$$M = h^x f^r \tag{2}$$

A variant of Schnorr protocol [10] with two responses (for  $x$  and for  $r$ ) can be used to show knowledge of values committed. This commitment scheme is computationally binding and information-theoretic hiding.

An *interactive proof system* [7] for a language  $L$  is an interactive pair of Turing machines with a word  $X$  on common input tape such that *completeness* and *soundness* holds. A party is *honest* if it follows the protocol. Completeness is honest Verifier always accepts for an honest Prover and a word from language. Soundness is honest Verifier accepts for any Prover and a word not from language with only a negligible probability. An *interactive argument* [3, 4] is a proof system with an additional auxiliary input tape for Prover with a *witness* to language membership on it, and with soundness depending on infeasibility for a polynomial Prover to solve a hard problem. A proof system is *of knowledge* [1] if an *extractor* algorithm exists that outputs the witness from Prover responses given oracle access to Prover. A proof system is *zero knowledge* if a *simulator* algorithm exists for Verifier that outputs a *simulated transcript* indistinguishable from a transcript of a protocol with a Prover for any word from language. A proof system is *honest verifier zero knowledge* [2] if simulated transcript is indistinguishable from all protocol transcripts having the same challenge. Protocols with *binary* challenges require repeating to achieve small probability of cheating, while protocols with challenges chosen from a set of a large cardinality achieve soundness in constant round.

We observe Prover can only produce acceptable responses of Schnorr protocol as estimates of a polynomial linear in challenge. We choose an appropriate *verification polynomial* to test language membership for values committed. Verification polynomial can be evaluated only using responses and challenges of Schnorr protocol.

Let  $F(z)$  be a non-zero polynomial of degree  $n$  over a ring. Consider a Verifier choosing a challenge  $c$  from a set  $S$ . Schwartz-Zippel lemma [11] says that probability to choose a root of such a polynomial (that is,  $F(c) = 0$ ) at random is at most  $\frac{n}{|S|}$ .

### 3 A protocol to show a codeword

Consider an equation equivalent to definition of Goppa codeword (1):

$$\sum_{j=1}^N b_j \sum_{k=1}^T g_k \frac{z^k - a_j^k}{z - a_j} \prod_{i \neq j} \sum_{m=0}^T g_m a_i^m \equiv 0 \quad (3)$$

$$g(z) = \sum_{k=0}^T g_k z^k \quad (4)$$

where  $\{b_j\}$  are codeword components, and  $\{g_k\}$  are coefficients of polynomial  $g(z)$ .

We verify that (3) holds by choosing some nonzero  $d \in \mathbb{F}_q$  at random as a challenge, and testing that it holds for  $z = d$ .

We consider a bivariate *verification polynomial* produced by substituting responses of Schnorr protocol in place of codeword components and coefficients:

$$\Gamma(z, y) = \sum_{j=1}^N (y b_j + \beta_j) \sum_{k=1}^T (y g_k + \alpha_k) \frac{z^k - a_j^k}{z - a_j} \prod_{i \neq j} \sum_{m=0}^T (y g_m + \alpha_m) a_i^m \quad (5)$$

Verification polynomial is of power  $N + 1$  in  $y$ . This polynomial is of power at most  $N$  if, and only if equation (3) holds for  $(\{b_j\}, \{g_k\})$ . We test that  $\Gamma(d, y)$  can be reproduced with exactly  $N$  coefficients for  $y = c$  chosen as another challenge. See protocol for multiplication [6] as a background.

Common input is group description, two group elements, support set and Pedersen commitments:  $X = (q, h, f, \{V_k\}, \{W_j\})$ . Auxiliary input of Prover is  $(\{g_k\}, \{\theta_k\}, \{b_j\}, \{\phi_j\})$  such that:

$$V_k = h^{g_k} f^{\theta_k}, \quad k = 0 \dots T \quad (6)$$

$$W_j = h^{b_j} f^{\phi_j}, \quad j = 1 \dots N \quad (7)$$

Prover shows that values committed constitute Goppa polynomial and a codeword with a protocol shown on Figure 1.

It can be shown that probability for an honest Verifier to accept for any polynomial Prover and for any data committed that do not satisfy Goppa codeword definition is at most  $\frac{T+N}{q}$  (over choices of Verifier).

### 4 A protocol to show a bounded error

Let  $\{w_j\}$  be a damaged codeword with an error  $\{e_j\}$ :  $w_j = b_j + e_j$ . Consider a Prover willing to show that  $|\{j \mid e_j \neq 0\}| \leq S$  for some  $S$  agreed

( $S \leq T$ ). Verifier tests that *error verification polynomial*

$$\Gamma'(y) = \prod_{j=1}^N (yb_j + \beta_j - yw_j) \quad (19)$$

is of power at most  $S$  in  $y$ . See protocol for set membership [6] as a background.

Common input is  $(q, h, f, \{w_j\}, \{W_j\})$ , auxiliary input of Prover is  $(\{b_j\}, \{\phi_j\})$  such that equations (6, 7) hold. Prover shows that

$$|\{j \mid w_j - b_j \neq 0\}| \leq S \quad (20)$$

for a codeword committed with a protocol shown on Figure 2.

It can be shown that probability for an honest Verifier to accept for any polynomial Prover and an error of Hamming weight of more than  $S$  is at most  $\frac{N}{q}$  (over choices of Verifier), on condition of taking logarithms is hard for Prover.

## 5 Properties of protocols

Consider a case that (1) does not hold for  $\{g_k\}, \{b_j\}$  committed. According to Schwartz-Zippel lemma, probability for a Verifier to choose some  $d \in \mathbb{F}_q$  such that (1) holds for  $z = d$  is at most  $\frac{T-1}{q}$ ; verification polynomial is of power  $N + 1$  otherwise. It follows  $-\Gamma(d, y) + \sum_{l=0}^N y^l r_l$  is a non-zero polynomial for any  $\{r_l\}$ . According to Schwartz-Zippel lemma, probability for an honest Verifier to choose some  $c \in \mathbb{F}_q$  such that  $\Gamma(d, c) = \sum_{l=0}^N c^l r_l$  for any  $\{r_l\}$  chosen by Prover is at most  $\frac{N+1}{q}$ .

Any Prover capable of producing a pair of responses  $(\Psi'_k, \Theta'_k)$  that pass (16) and are not estimates of linear polynomials

$$\Psi'_k \neq \Psi_k(c), \quad \Psi_k(y) = yg_k + \alpha_k \quad (28)$$

$$\Theta'_k \neq \Theta_k(c), \quad \Theta_k(y) = y\theta_k + \zeta_k \quad (29)$$

is also capable to solve for  $\log_h(g)$ . The same holds for  $(\Omega_j, \Phi_j)$  and (17). We conclude protocol introduced achieve negligible soundness error  $\frac{N+T}{q}$  without repeating.

Consider a case with an error  $\{e_j\}$  of more than  $S$  weight. It follows  $-\Gamma'(y) + \sum_{l=0}^S y^l p_l$  is non-zero for any choice of  $\{p_l\}$ . According to Schwartz-Zippel lemma, probability for an honest Verifier to choose some  $c \in \mathbb{F}_q$  such that  $\Gamma'(c) = \sum_{l=0}^S c^l p_l$  for any  $\{p_l\}$  chosen by Prover is at most  $\frac{N}{q}$  (over random coins of Verifier).

Extractor algorithm for both protocols introduced is the same as one for Schnorr protocol (rewinding procedure). Namely, Prover is requested

to produce two sets responses to two different challenges without choosing another set of initial random coins.

Consider a simulator candidate algorithm for 'codeword' protocol. Given challenges  $(c, d)$ , Verifier chooses some field elements for responses  $\{\Psi_k\}, \{\Omega_j\}, \{\Theta_k\}, \{\Phi_j\}, \Delta$  uniformly at random, and some group elements for  $R_l, l = 1 \dots N$  uniformly at random. Verifier produces  $\Gamma$  according to (15). Verifier produces

$$U_k = h^{\Psi_k} f^{\Theta_k} V_k^{-c} \quad Q_j = h^{\Omega_j} h^{\Phi_j} W_j^{-c} \quad R_0 = h^\Gamma f^\Delta \prod_{l=1}^N (R_l)^{-c^l} \quad (30)$$

Simulated transcript is  $(d, \{U_k\}, \{Q_j\}, \{R_l\}, c, \{\Psi_k\}, \{\Omega_j\}, \{\Theta_k\}, \{\Phi_j\}, \Delta)$ .

It is clear that distribution of transcript components is flat and is identical to that of any transcript with a Prover with the same challenges  $d, c$ .

Consider a simulator candidate algorithm for 'bounded error' protocol. Given a challenge  $c$ , Verifier chooses some field elements for responses  $\{\Omega_j\}, \{\Phi_j\}, \Delta'$  uniformly at random and some group elements for  $P_l, l = 1 \dots S$  uniformly at random. Verifier produces  $\Gamma'$  according to (25). Verifier produces

$$Q_j = h^{\Omega_j} h^{\Phi_j} W_j^{-c} \quad P_0 = h^{\Gamma'} f^{\Delta'} \prod_{l=1}^S (P_l)^{-c^l} \quad (31)$$

Simulated transcript is  $(\{Q_j\}, \{P_l\}, c, \{\Omega_j\}, \{\Phi_j\}, \Delta')$ .

It is clear that distribution of transcript components is flat and is identical to that of any transcript with a Prover with the same challenge  $c$ .

## 6 Discussion

Protocols introduced can be useful for watermarking.

## References

- [1] Mihir Bellare and Oded Goldreich. On defining proofs of knowledge. In *CRYPTO*, pages 390–420, 1992.
- [2] Mihir Bellare, Silvio Micali, and Rafail Ostrovsky. The (true) complexity of statistical zero knowledge. In *STOC*, pages 494–502, 1990.
- [3] Gilles Brassard, David Chaum, and Claude Crépeau. Minimum disclosure proofs of knowledge. *J. Comput. Syst. Sci.*, 37(2):156–189, 1988.

- [4] Gilles Brassard, Claude Crépeau, and Moti Yung. Everything in  $np$  can be argued in perfect zero-knowledge in a bounded number of rounds. In *ICALP*, pages 123–136, 1989.
- [5] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *CRYPTO*, pages 174–187, 1994.
- [6] Vadym Fedyukovych. Proving polynomial identities (a presentation in Russian). In *Information Security conference, Kiev*, 2008. Presentation available.
- [7] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. Comput.*, 18(1):186–208, 1989.
- [8] F. J. MacWilliams and N. J. A. Sloane. *The theory of error-correcting codes*. North-Holland, 1977.
- [9] Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *CRYPTO*, pages 129–140, 1991.
- [10] Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In *CRYPTO*, pages 239–252, 1989.
- [11] J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, 1980.

Common input:  $(q, h, f, \{a_j\}, \{V_k\}, \{W_j\})$ .

Prover input:  $(\{g_k\}, \{\theta_k\}, \{b_j\}, \{\phi_j\})$  such that  $V_k = h^{g_k} f^{\theta_k}$   $W_j = h^{b_j} f^{\phi_j}$

Prover shows that  $\sum_{j=1}^N \frac{b_j}{z-a_j} \equiv 0 \pmod{g(z)}$  for  $g(z) = \sum_{k=0}^T g_k z^k$

1. Verifier chooses a non-zero challenge  $d \in \mathbb{F}_q$  at random, and sends it to Prover.
2. Prover chooses  $(\alpha_k, \zeta_k) \in \mathbb{F}_q^2$ ,  $(\beta_j, \eta_j) \in \mathbb{F}_q^2$ ,  $\mu_l \in \mathbb{F}_q$  at random, produces initial commitments  $\{U_k\}, \{Q_j\}$ , expansion coefficients  $\{r_l\}$ , commitments  $\{R_l\}$ :

$$U_k = h^{\alpha_k} f^{\zeta_k}, \quad k = 0 \dots T \quad (8)$$

$$Q_j = h^{\beta_j} f^{\eta_j}, \quad j = 0 \dots N \quad (9)$$

$$\sum_{j=1}^N (y b_j + \beta_j) \sum_{k=1}^T (y g_k + \alpha_k) \frac{d^k - a_j^k}{d - a_j} \prod_{i \neq j} \sum_{m=0}^T (y g_m + \alpha_m) a_i^m = \sum_{l=0}^N y^l r_l \quad (10)$$

$$R_l = h^{r_l} h^{\mu_l}, \quad l = 0 \dots N \quad (11)$$

Prover sends  $(\{U_k\}, \{Q_j\}, \{R_l\})$  to Verifier.

3. Verifier chooses his second non-zero challenge  $c \in \mathbb{F}_q$  at random, and sends it to Prover.
4. Prover produces responses  $(\{\Psi_k\}, \{\Theta_k\}, \{\Omega_j\}, \{\Phi_j\}, \Delta)$  and sends them to Verifier

$$\Psi_k = c g_k + \alpha_k \quad \Theta_k = c \theta_k + \zeta_k \quad (12)$$

$$\Omega_j = c b_j + \beta_j \quad \Phi_j = c \phi_j + \eta_j \quad (13)$$

$$\Delta = \sum_{l=0}^N c^l \mu_l \quad (14)$$

5. Verifier produces

$$\Gamma = \sum_{j=1}^N \Omega_j \sum_{k=1}^T \Psi_k \frac{d^k - a_j^k}{d - a_j} \prod_{i \neq j} \sum_{m=0}^T \Psi_m a_i^m \quad (15)$$

Verifier accepts if

$$h^{\Psi_k} f^{\Theta_k} V_k^{-c} = U_k \quad (16)$$

$$h^{\Omega_j} h^{\Phi_j} W_j^{-c} = Q_j \quad (17)$$

$$h^{-\Gamma} f^{-\Delta} \prod_{l=0}^N (R_l)^{c^l} = 1 \quad (18)$$

Figure 1: Protocol for codeword of Goppa code

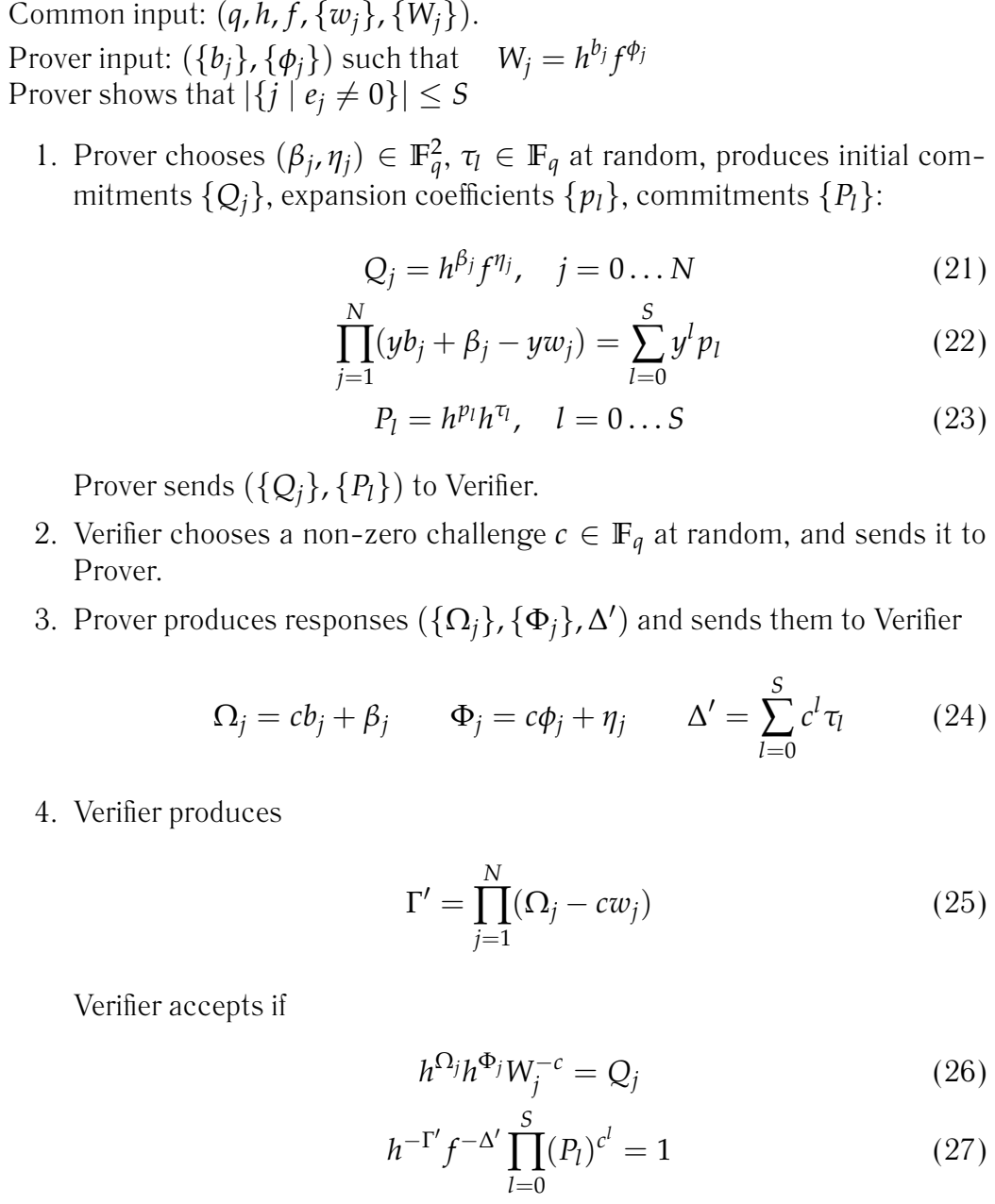


Figure 2: Protocol for bounded error