The Jacobi symbol is a generalisation of the Legendre symbol to integers n which are odd but not necessarily prime. Let $n \geq 3$ be odd with prime factorisation $n = p_1^{e1} p_2^{e2} \ldots p_k^{ek}$. Then the Jacobi symbol (a/n) is defined to be

$$\left(\frac{a}{n}\right) = \left(\frac{a}{p_1}\right)^{e_1} \left(\frac{a}{p_2}\right)^{e_2} \cdots \left(\frac{a}{p_k}\right)^{e_k}$$

Let $n \geq 3$ be an odd integer, and $J_n = \{a \in \mathbb{Z}_n^* \mid (a/n) = 1\}$. The set of pseudosquares modulo n, denoted $\tilde{Q}_n$, is defined to be the set $J_n - Q_n$. Then $|Q_n| = |\tilde{Q}_n| = (p-1)(q-1)/4$; that is, half of the elements in J_n are quadratic residues and the other half are pseudosquares.

A Blum integer is a composite integer of the form $n = pq$, where p and q are distinct primes each congruent to 3 modulo 4. Let $n = pq$ be a Blum integer, and let $a \in J_n$, then

$$a^{2d} \equiv \begin{cases} a & \text{if } a \in Q_n \\ n - a & \text{if } a \in \tilde{Q}_n \end{cases}$$

where $d = (n - p - q + 5)/8$. In particular, if $p \equiv 3 \pmod 8$ and $q \equiv 7 \pmod 8$ (such an integer n is sometimes called a Williams integer), then 2 is a quadratic non-residue modulo n having Jacobi symbol $(2/n) = -1$. Then, for each $a \in \mathbb{Z}_n^*$ we have $a \in J_n$ or $2a \in J_n$ [3].

Identification scheme: In this Section, a new interactive identification scheme is proposed, with which the prover A can prove its identity to verifier B in t executions of a three-pass protocol.

(i) *Selection of system parameters:* Each user is given a Williams integer $n = pq$. He publishes n as the public parameter and keeps primes p and q secret. In addition, let k be an integer such that $2^k < \min\{p, q\}$.

(ii) *Protocol messages:* Each of t rounds has three messages with form as follows:

$$A \rightarrow B : x = r^2 \bmod n \tag{1}$$

$$A \leftarrow B : y = 2^b \quad b \in \{0, 1\} \quad 0 < s < 2^k \tag{2}$$

$$A \rightarrow B : z = ry^d \quad d = (n - p - q + 5)/8 \tag{3}$$

(iii) *Protocol actions:* The following steps are iterated t times (sequentially and independently). B accepts the proof if all t rounds succeed.

(*a*) A chooses a random (commitment) $r \in \mathbb{Z}_n^*$, and sends (the witness) $x = r^2 \bmod n$ to B.

(*b*) B randomly selects a k bit (challenge) s ($1 \leq s < 2^k$), and sends $y = 2^b s$, where $b = 0$ if $s \in J_n$ and $b = 1$ if $2s \in J_n$.

(*c*) A computes and sends to B (the response) $z = ry^{(n-p-q+5)/8}$.

(*d*) B accepts the proof if $z^2 = \pm xy \bmod n$, and rejects otherwise.

In the protocol the interactive procedure requires only four modular multiplications, computing one Jacobi symbol and one exponentiation. In terms of storage requirements and communication, the new scheme is superior to other schemes.

Security of scheme: It is well-known that computing square roots modulo n is difficult when n is a large composite integer for which the prime factors are unknown. In fact it is equivalent to factoring n [3]. The security of the scheme relies on the difficulty of computing square roots modulo a composite n, and on the following:

(i) *Completeness:* If the prover and the verifier follow the protocol, then the verifier always accepts the proof as valid.

(ii) *Probability of forgery:* Similar to [4], we can prove that the prover does not know the primes p and q, then following the protocol the verifier will accept the proof as valid with probability bounded by 2^{-k}.

(iii) *Soundness:* Since commitment r is randomly selected, the adversary does not have any chance to obtain essential distinct square roots of the response using a chosen-text attack, in order to extract the factoring of $n = pq$. In fact, if different roots r_1 and r_2 are known, a non-trivial factor of n can be found by computing $gcd(r_1 \pm r_2, n)$. If the prover knows the quadratic roots of the random number, he is believed likely to know the factoring of $n = pq$.

(iv) *Zero-knowledge:* The protocol can be simulated for k not large, e.g. $k = O(\log \log n)$, because there is an expected polynomial-time simulator which can produce transcripts indistinguishable from those resulting from interaction with the real prover.

(v) *Parameter selection:* Choosing k and t such that $kt > 20$ allows a $1/10^7$ chance of impersonation, which is sufficient in the case where an identification attempt requires a personal appearance by a would-be impersonator. Computation, memory, and communication can be traded off for different k and t. Specific parameter choices might be, for security 2^{-20} : $k = 5$, $t = 4$ or $k = 10$, $t = 2$.

(vi) *Security tradeoff:* Both the computational and communication burdens may be reduced by trading off security parameters to yield a single iteration ($t = 1$), holding the product kt constant and increasing k while decreasing t; however, in this case the protocol is no longer a zero-knowledge proof of knowledge.

Conclusions: A new interactive identification protocol has been presented which is based on the quadratic residue (QR), has low computational and communication overheads and only requires a small amount of memory in which to store the secret information. In comparison with another scheme based on the QR [4], the new scheme achieves the same security level, but is simpler and more efficient. Owing to simplicity, security and speed, the proposed scheme can be applied to microprocessor-based devices such as smart cards etc.

Acknowledgments: I wish to thank the support of the APA program financed by the German Ministry for Development and Economic Cooperation. This work has been supported by NSFC under the grant 69773013.

8 September 1998
Electronics Letters Online No: 19981555

Kefei Chen (*Department of Computer Science and Engineering, Shanghai Jiaotong University, Shanghai 200030, People's Republic of China*)

E-mail: chen-kf@cs.sjtu.edu.cn

References

1 FEIGE, U., FIAT, A., and SHAMIR, A.: 'Zero-knowledge proofs of identity', *J. Cryptol.*, 1988, **1**, (2), pp. 77–94
2 GUILLOU, L.C., and QUISQUATER, J.J.: 'A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory'. Proc. Eurocrypt '88, 1988, Paper LNCS 330, pp. 123–128
3 MENEZES, A.J., VAN OORSCHOT, P., and VANSTONE, S.A.: 'Handbook of applied cryptography' (CRC Press, Boca Raton, 1997)
4 NYANG, D., KIM, E., and SONG, J.: 'An interactive identification scheme based on quadratic residue problem', *IEICE Trans. Fundamentals*, 1997, **E80-A**, (7), pp. 1330–1335
5 SCHNORR, C.P.: 'Efficient signature generation by smart cards', *J. Cryptology*, 1991, **4**, (3), pp. 161–174

Method for hiding information in lattice

DaeHun Nyang and JooSeok Song

A method is presented for encoding a message in the lattice of the GGH public-key cryptosystem. The approach is more efficient than the authors' original contribution and also has a stronger notion of plaintext awareness. With the encoding scheme, encryption of a given message has a computational requirement of only $O(n)$, which is superior to the original scheme which requires $O(n^2)$.

Introduction and motivation: It has been found that the closest vector problem (CVP) and the shortest vector problem (SVP) may be useful in various cryptographic protocols. The CVP was shown by van Emde Boas to be NP-hard in 1981 [4]. In 1996, Ajtai introduced a function that is probably one-way if it is difficult to approximate the shortest nonzero vector (SVP) in a lattice, and invented a public-key cryptosystem using the lattice problem [2, 3].

Being motivated by Ajtai's work, Goldreich, Goldwasser, and Halevi proposed public-key cryptosystems using lattice reduction problems [1]. The security of the GGH cryptosystem depends on the SVP and CVP. When Alice wants to encrypt a message with Bob's public-key, she encodes the message in a lattice point and

then adds noise to it. The resultant vector is a ciphertext. Mallory must solve the CVP if she wants to decrypt the ciphertext only with Bob's public-key. Bob's public-key B has a large orthogonality defect even after LLL reduction, where the orthogonality defect is defined as orth-defect $(B) = \Pi_i \|b_i\|/det(B)$ and $\|b_i\|$ is the Euclidean norrn of b_i. In contrast to the public-key, the corresponding private-key R has a sufficiently small orthogonality defect. With R, Bob can easily recover the original message. If Mallory tries to recover R from B, she must solve the SVP.

When a message is encoded as vectors in Z^n, we must check that the encoding function reveals almost no information about the message even if a few entries in the encoded vector v are exactly known or some rough estimates of all the other entries of v are known. In the GGH scheme, two candidates for the encoding function are proposed. One is estimated using the trapdoor one-way function, and the other using the low-order bits in v. With these encoding schemes, the GGH cryptosystem has a computationsl requirement of $O(n^2)$ both for encryption and decryption.

We propose a method for encoding a message which has greater secrecy, and which also speeds up the encryption. The idea behind it is the observation that a message can be embedded in the error vector. The vector v is chosen randomly from R^n and independently of the message to be encrypted. Because v is independent of the message, Alice can precompute the main time-consuming part, which requires $O(n^2)$ operations. When a message to be encrypted is given, all that Alice needs to do is encode the message and add it to the precomputed vector. Both require only $O(n)$ operations.

Our proposal: First, a lattice is defined.

Definition 1 (lattice): Given a set of n linearly independent vectors in R^n, $B = \{b_1, ..., b_n\}$, we define the lattice spanned by the basis B as the set of all linear combinations of the b_is with integral coefficients, i.e.

$$L(B) = \left\{ \sum_i k_i b_i : k_i \in Z \text{ for all } i \right\}$$

Definition 2 (dual lattice): Let $B = b_1, ..., b_n$ be a basis for some lattice in R^n, $L = L(B)$. The dual lattice of L is the lattice that is spanned by the rows of matrix B^{-1}.

We use the same trapdoor function as that of the GGH scheme. The public-key and private-key consist of (B, σ) and (R, T), respectively, where T is the unimodular transformation matrix $T = B^{-1} R$. Our modification of the GGH scheme is as follows:

(i) *Encryption*:

(1) *Encoding a message $m \in \{0, 1\}^{n/2}$ in an error vector*:

$$\text{For } 1 \le i \le n/2, \quad s_i = -\sigma \text{ if } m_i = 0$$
$$\sigma \text{ if } m_i = 1$$

$t = -Ps$, where P is a random permutation matrix. $e^T = s^T \| t^T$, where x^T is the transpose of vector x.

(2) *Adding the error vector to a lattice point*: Find a random lattice point $v \in L$. Calculate the following: $c = Bv + e$, where the computation of Bv is carried out in a preprocessing stage.

(ii) *Decryption* :

(1) *Calculating the error vector*:

$$v = T\lceil R^{-1} c \rfloor$$
$$e = c - v$$

$\lceil c \rfloor$ maps the nearest integer of c.

(2) *Decoding*:

$$\text{For } 1 \le i \le n/2, \quad m_i = 0 \text{ if } e_i = -\sigma$$
$$1 \text{ if } e_i = \sigma$$

The key generation procedure is the same as that of the GGH scheme. Private-key R has a low dual-orthogonality-defect and public-key B has a high dual-orthogonality-defect. Both are represented by $n \times n$ matrices and span the same lattice L. There is another public-key entry σ, which is used originally for just error vector generation, but is now used for message encoding. For the private basis R, σ should be less than $1/(2\rho)$ if no decryption errors can occur, where ρ is the maximum L_1 norm of the rows in R^{-1}.

The reason that we concatenate the permuted and negated vector of a message to the message vector is to obtain an average of the error vector zero. This process thwarts the brute-force attack at the expense of storage efficiency. Without the concatenation process, the average of an error vector e will not be zero and the problem space Mallory must search is not bounded by $(\pi e)^{n/2} \cdot \sigma^n \cdot \text{orth-defect}^*(B)/det(B)$, where the orth-defect$^*(B)$ is the orthogonality defect of the dual lattice of $L(B)$.

Security: The security of our scheme is equivalent to that of the GGH scheme's setting. Assume that there is a polynomial time algorithm A that breaks our scheme. Then, A can be used to break the GGH scheme in polynomial time.

(a) Using A, compute e that includes the message vector for a given c.

(b) With e and c, compute v by $c - e$.

(c) Output v.

The above algorithm computes v that is the closest lattice point to c in polynomial time, since step b requires $O(n)$ computations. Thus, our scheme is as secure as the GGH scheme.

In our scheme, we do not need to consider the security holes of the encoding scheme, which are referred to in [1]. Our encoding scheme ensures that knowing a few entries in v exactly and knowing some rough estimates of all the other entries does not help Mallory recover the original message, owing to the randomness of v.

Without using the trapdoor function paradigm in the GGH scheme, we achieve polynomial time indistinguishability between any two messages by masking a message with a randomly selected lattice point.

Probabilistic encryption and plaintext awareness: Whenever we encrypt a message, the corresponding ciphertext differs. This 'probabilistic' nature of the cryptosystem is desirable in that it is highly resistant to the chosen plaintext attack, which is generally the weakest point of a public-key cryptosystem. The concept of probabilistic cryptosystems was first introduced in [5].

In the GGH scheme there is a weak notion of 'plaintext awareness'. Our system, however, has a stronger notion of plaintext awareness. Even though a valid ciphertext c is known, Mallory cannot generate another valid ciphertext. Thus, our scheme is resistant to adaptive chosen-ciphertext attacks. Unlike in the GGH scheme, adding any lattice point to c does not generate any different ciphertext such that the respective plaintexts are related in a known manner. Any ciphertext obtained in this way is merely a valid ciphertext of the same message. Adding another error vector, $e' \in \{2\sigma, 0, -2\sigma\}^n$ to c generates an invalid ciphertext with high probability. Here, 'invalid ciphertext' means ciphertext that cannot recover the original lattice point v. Mallory chooses e' from the set $\{2\sigma, 0, -2\sigma\}^n$. If she guesses that the message bit is σ and wants to change the bit, she will choose -2σ and add it to c. If she does not want to change the bit, she will choose 0 and vice versa.

We define the inversion error probability as P_{inv}. We now derive P_{inv}. We denote $d = R^{-1}(e + e')$ and the ith entry in d and $e + e'$ by δ_i and ε_i, respectively. Also, we denote the maximum L_1 norm of the rows in R^{-1} by ρ. For the inversion error probability to be zero, $\lceil R^{-1}(e + e') \rfloor$ should be zero, and this leads to $|\delta_i| \le \sigma \cdot \rho < 1/2$ for every i. Thus, if σ is chosen such that $\sigma < 1/(2\rho)$, no inversion error can occur. Thus, if any entry in the error vector is greater than the chosen σ, it will cause an inversion error.

When Mallory correctly guesses the message bit, she will not cause the inversion error because ε_i is either σ or $-\sigma$. However, when she guesses incorrectly and wants to change the bit, ε_i becomes either 2σ or -2σ and causes an inversion error. Thus, $P_{inv} = 1 - (3/4)^n$, if we assume that Mallory wants to change a message bit with the same probability as leaving it untouched. If we work in dimension 200, $P_{inv} \simeq 1 - 1.028 \times 10^{-25}$. The probability of an inversion error is bounded by $\Pr[|\delta_i| > 1/2] < 2\exp(-1/32\sigma^2\gamma^2)$, which is much higher than that of the GGH scheme, where $\gamma\sqrt{n}$ is the maximum L_∞ norm of the rows in R^{-1}.

Performance: It takes only $O(n)$ operatives to encrypt a message with our scheme, in contrast to the $O(n^2)$ required in the GGH scheme. This is entirely due to the precomputable structure of our scheme. That is, before being given a message to encrypt, we can

prepare a random lattice point v and precompute Bv in our idle time. When a message is given to be encrypted, only the encoding and addition of Bv and e are required, where Bv is already computed. The encoding of a message involves a simple translation of a binary number into $\pm\sigma$ and a random permutation. The latter can be processed in $O(n)$, although the above description allows for the inclusion of a multiplication of an $n \times n$ matrix and vector. Thus, the entire running time is $O(n)$.

In the original contribution, the authors suggest two encoding schemes. One has low bandwidth utilisation: it can send only $\log n$ bits of messages at a time. Our scheme is superior to that, and can send messages at a rate of $n/2$ bits.

Generalisation hint: The error vector may be quantised to a more delicate level. This multilevel quantisation permits a high information rate. That is, a single ciphertext contains much more information than in the case of two level quantisation. For an n dimensional lattice vector, $n/2$ bit data are encoded with two-level quantisation. With q-level quantisation, $n/2 \lg q$ bits of data will be encoded. However, multilevel quantisation requires multi-precision handling and it may cause decoding errors. Thus, special care must be taken when a multi-precision quantisation technique is used for analogue data encryption.

Conclusion: In this Letter, we have presented a method for encoding a message with a lattice-based public-key cryptosystem. The idea behind it is to embed a message in an error vector rather than in a lattice vector. The encoding scheme use quite a fast encryption procedure and has plaintext awareness. Only $O(n)$ operations are required to encrypt a message, which is a significant enhancement compared to the $O(n^2)$ of the original GGH scheme, and the $O(n^3)$ of RSA. Also, we have briefly described a multilevel quantisation technique for improving the information rate.

7 October 1998
Electronics Letters Online No: 19981589

DaeHun Nyang and JooSeok Song (*Department of Computer Science, Yonsei University, SeodaemunGu, ShinchonDong 134, Seoul 120-749, Korea*)

References

1 GOLDREICH, O., GOLDWASSER, S., and HALVEI, S.: 'Public-key cryptosystems from lattice reduction problems'. Proc. CRYPTO'97, Santa Barbara, CA,1997, pp. 112–131
2 AJTAI, M.: 'Generating hard instances of lattice problems'. Proc. 28th STOC, Philadelphia, 1996, pp. 99–108
3 AJTAI, M., and DWORK, C.: 'A public-key cryptosystem with worst-case/average-case equivalence'. Proc. 29th STOC, Texas, 1997, pp. 284–293
4 VAN EMDE BOAS, P.: 'Another NP-complete problem and the complexity of computing short vectors in a lattice'. Report 81-04, Mathematische Instituut, University of Amsterdam, 1981
5 GOLDWASSER, S., and MICALI, S.: 'Probabilistic encryption', *J. Comput. Syst. Sci.*, 1984, **28**, pp. 270–299

Modified turbo codes with low decoding complexity

Li Ping

A family of modified turbo codes with SPC precoding is presented. Compared with standard turbo codes, the new scheme results in considerably lower decoding cost yet can achieve nearly the same performance.

Introduction: In this Letter, a modified turbo code scheme is presented. The basic principle is to use SPC (single parity check) precoding to replace the puncturing technique [1] for rate adjustment. This greatly reduces the length of the convolutional code involved. It is also proposed that very simple constituent convolutional codes be used, e.g. those with four states, and that the number of constituent codes (so-called dimensions) be increased [2, 3] to maintain performance. Compared with the standard turbo codes [1], the new scheme has considerably lower decoding cost yet can achieve nearly the same performance. The method is convenient for constructing median to high rate codes.

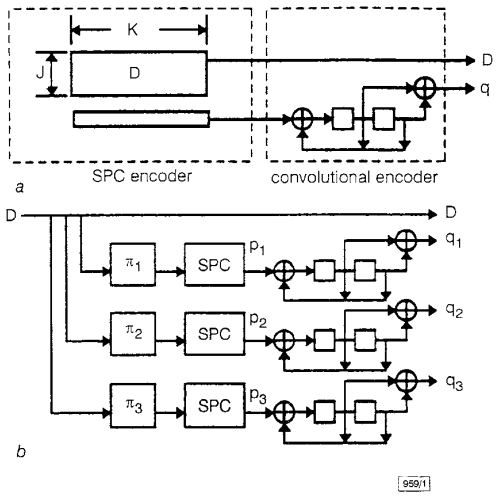

Fig. 1 *SPC-convolutional encoder and three-dimensional SPC-turbo code*

a SPC-convolutional encoder
b Three-dimensional SPC-turbo code
$\{\pi_n\}$ are interleavers and SPC stands for SPC encoder in Fig. 1*a*

Encoding and decoding principles: Fig. 1*a* illustrates the encoding principle of the constituent code. The information bits are arranged as a $J \times K$ array $\mathbf{D} = (D_{j,k})$. For convenience we will assume that $\mathbf{D}$ is represented in the binary phase shift keying (BPSK) format over $\{+1, -1\}$. A parity check p_k is generated for every column of $\mathbf{D}$ so that $\mathbf{D}_k \cup p_k$ contain an even number of -1. The resultant vector $\mathbf{p} = \{p_k\}$ is used to drive a convolutional encoder with output sequence $\mathbf{q}$.

A three-dimensional concatenated code is illustrated in Fig. 1*b* based on that in Fig. 1*a*. It will be referred to as an SPC-turbo code. The information array $\mathbf{D}$ is interleaved and encoded three times, producing $\mathbf{q}_1$, $\mathbf{q}_2$ and $\mathbf{q}_3$. The overall codeword is formed by $(\mathbf{D}, \mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3)$, with rate $R = J/(J + 3)$. The principle can be generalised to N dimensions.

Decoding method: The global iterative decoder structure in [3] is adopted here for general multidimensional concatenated codes, which requires the MAP (maximum *a posteriori*) decoder for the constituent code. This can be accomplished for the code in Fig. 1*a* by treating it as a rate $J/(J + 1)$ convolutional code, but the related trellis description has a high branch complexity (with 2^J branches emanating from every node). A more efficient method is now outlined. Only one dimension will be considered here so the dimension subscript for $\mathbf{q}$ will be omitted.

Let $obs(\mathbf{c})$ be the noisy observation of a transmitted codeword $\mathbf{c}$ in the BPSK format. Define the *a priori* LR (likelihood ratio) for a single bit c_n conditioned on $obs(c_n)$ as

$$\tilde{L}(c_n) \triangleq \frac{\Pr\{c_n = +1 | obs(c_n)\}}{\Pr\{c_n = -1 | obs(c_n)\}} \qquad (1)$$

This is equivalent to the definition of LLR (logarithm of likelihood ratio) [1, 4]. Assuming that the *a priori* LR values are available for all the bits, the aim of the MAP decoding is to find the *a posteriori* LR for c_n as

$$L(c_n | obs(\mathbf{c})) \triangleq \frac{\Pr\{c_n = +1 | obs(\mathbf{c})\}}{\Pr\{c_n = -1 | obs(\mathbf{c})\}} \qquad (2)$$

Eqn. 2 is evaluated for the code in Fig. 1*a* using the three step technique below. It is an exact solution but the proof will not be included due to space limitations.

Step (i): Let $\mathbf{D}_k = \{D_{j,k}\}$ be the k-column of $\mathbf{D}$. Compute $L(p_k | obs(\mathbf{D}_k))$ for p_k, $k = 1, 2 \ldots K$.

Step (ii): Use $L(p_k | obs(\mathbf{D}_k))$, $k = 1, 2, \ldots K$, as the *a priori* LR for $\mathbf{p}$. Apply MAP decoding to the rate 1/2 convolutional code $\mathbf{p} \rightarrow (\mathbf{p}, \mathbf{q})$. This produces $L(p_k | obs(\mathbf{D}, \mathbf{q}))$. Define the extrinsic LR as $W(p_k) = L(p_k | obs(\mathbf{D}, \mathbf{q}))/L(p_k | obs(\mathbf{D}_k))$.