A Multiple-Iterated Trapdoor for Dense Compact Knapsacks

Glenn Orton

Queen's University / Cryptalis Data Security

Abstract–**A modification to the multiple-iterated Merkle–Hellman trapdoor is described that permits a knapsack density exceeding the critical density 0.94 of the Lagarias–Odlyzko low-density attack. A high density level also permits fast signature generation. Compaction and common knapsack weights are used to reduce the public-key size. The security of the new trapdoor depends on a simultaneous diophantine approximation problem plus a residue recombination problem.**

1 Introduction

Shamir [16] found that trapdoor knapsacks are not necessarily as hard as the worst case knapsacks studied by complexity theory [10]. Desmedt, Vanderwalle, and Govaerts [20] defined the "useless" knapsacks as the set of knapsack problems that can not be generated by any number of Merkle–Hellman (MH) trapdoor iterations [6]. These suspicions were later confirmed when many knapsack public-key cryptosystems were found to be susceptible to polynomial-time low-density attacks [30]–[38] although the knapsack problem is NP-complete [10]. The Lagarias–Odlyzko low-density attack can solve "almost all" knapsacks of density below 0.94 [36],[37] (density is defined in §2) if the shortest vector in a lattice can be found. Finding the shortest vector is also an NP-complete problem [26]. Polynomial-time algorithms [27]–[29] exist to find relatively short vectors but their probability of success decreases with the problem size.

Knapsacks of density above 1.0 are not generally uniquely decodable. The Chor–Rivest knapsack [11] can achieve a density of about 1.3 because the message is expanded with the Bose–Chowla algorithm. The maximum density of a MH knapsack is $ns/(ns + (r-1)(s + \log n)) \leq 1$, where n is the number of knapsack variables of s bits each and r is the number of iterations of modular multiplication in the trapdoor (all logarithms will be base two). The MH knapsack density falls because their knapsack weights expand with multiple iterations. We have found a method of iterating a knapsack trapdoor without substantial weight expansion. We also expand the message by $r-1$ variables. Then the maximum density *increases* with the number of iterations to $1 + (r-1)(s + \log n)/ns \geq 1$.

Merkle and Hellman [6] proposed signing messages with their knapsack but the average number of signature generation attempts (i.e. message decodings) had a lower bound of $(ns)^{r-1}$. The average number of signature generation attempts for the new knapsack can approach 1.0. A knapsack is compacted by lowering n with ns constant because the public-key size is close to $n^2 s$ bits. The new knapsack is compactible and a subset of the knapsack weights may be common to all parties. This permits a public-key size of several kilobits considering known compact knapsack attacks [39]–[44].

The MH cryptosystem and variants such as the Graham–Shamir knapsack (see [17] for a description) and Goodman–McCauley knapsack [7] were also vulnerable to attacks on the

The research reported here was part of a Ph.D. program at Queen's University, Kingston, Ontario, Canada. The author is presently with Cryptalis Data Security, 109 Amelia St., Toronto, Ontario, M4X 1E5, (416) 927-7732, ORTON@FSW.Utoronto.CA.

trapdoor. The modular multiplications that disguised an "easy" knapsack leaked sufficient information for an adversary to unravel them [16]–[24]. As a solution to this weakness, Laih et. al [15] proposed to "linear shift" each knapsack weight by a random multiple of a constant after the trapdoor modular multiplications but this slows down decoding, especially with compaction. Previously proposed knapsack cryptosystems [12]–[15] that have withstood cryptanalysis are slower for decoding than the MH knapsack or can not be compacted.

We propose to publish a knapsack group consisting of congruence classes of any trapdoor knapsack. This knapsack group is a residue number representation of the original knapsack, hence the name: *residue knapsack*. Finding the trapdoor may well require that the original knapsack weights be recovered and that implies recombining the residues of the knapsack weights without knowing the moduli that uniquely define the congruence classes. Asmuth and Bloom's [9] secret sharing scheme is based on a similar residue recombination problem. Residue representation has a very small effect on the encoding and decoding speed and the information rate (as defined in §2) of a knapsack.

The MH knapsack is reviewed in §2. Readers familiar with the MH knapsack may skip to §3 where the residue knapsack is introduced. The security of the residue knapsack is analyzed in §4.

2 Background

2.1 Merkle and Hellman Knapsack

The "knapsack" or subset-sum problem is to find a combination of weights $\{a_1, a_2, \ldots, a_n\}$ that sums to a target value y:

$$y = \sum_{i=1}^{n} x_i a_i . \tag{1}$$

The solution to the knapsack problem is represented by $\{x_1, x_2, \ldots, x_n\}$ and x_i may be restricted to a range of values such as $[0, 2^s)$, where s is a positive integer. A knapsack's density is defined as

$$\frac{ns}{\log_2 A}, \tag{2}$$

where $A = \max\{a_1, a_2, \ldots, a_n\}$ (i.e. the largest weight). The information rate is ns/logy.

A message-to-be-encrypted is assigned to $\{x_1, x_2, \ldots, x_n\}$ and the ciphertext y is computed according to (1). To construct a public-key cryptosystem, a set of "easy" knapsack weights are translated to a set of "hard" knapsack weights, forming the public-key. Knowledge of the reverse transformation or "trapdoor" permits easy decryption.

The MH [6] "easy" knapsack weights are a superincreasing series a_i^0, for $i = 1$ to n, such as $\{1, 3, 7, 17, 35, \ldots\}$ assuming $s = 1$, where

$$a_i^0 > (2^s - 1) \sum_{j=1}^{i-1} a_j^0 . \tag{3}$$

Their disguising technique is modular multiplication by a constant. An r-times iterated trapdoor has final published weights of $a_i \triangleq a_i^r$, for $i = 1$ to n and initial weights a_i^0, where

$$a_i^k \equiv w^k a_i^{k-1} \bmod p^k, \tag{4}$$

for $k = 1$ to r, and $\gcd(p^k, p^{k+1}) = 1$. Unique decoding is ensured by the small-sum principle:

$$p^k > (2^s - 1) \sum_{i=1}^{n} a_i^{k-1}, \tag{5}$$

for $k = 1$ to r. The multiplicative constants w^k satisfy $\gcd(w^k, p^k) = 1$.

Decryption is performed by solving an "easy" knapsack problem with superincreasing weights a_i^0 and target value y^0, where $y^r \triangleq y$,

$$y^{k-1} \equiv y^k (w^k)^{-1} \bmod p^k, \tag{6}$$

and

$$x_i = \left\lfloor \frac{y^0 - \sum_{j=i+1}^{n} x_j a_j^0}{a_i^0} \right\rfloor, \tag{7}$$

for $i = n$ decrementing to 1.

2.2. Cryptanalysis of the Merkle and Hellman trapdoor

Shamir [21] found the first successful attack on the single-iterated MH knapsack. Adleman [23] found a feasible attack on the Graham–Shamir trapdoor (see [17] for a description) and was the first to employ the Lovasz lattice basis reduction algorithm [27]. Brickell [24] was the first to convincingly demonstrate a feasible attack on the multiple-iterated MH trapdoor.

Brickell's attack searches for an alternate trapdoor, defined by $b_i^{k-1} \equiv U^k b_i^k \bmod V^k$, where

$$b_i^{k-1} = U^k b_i^k - h_i^k V^k < V^k 2^{s+\log n}, \tag{8}$$

$b_i^r = a_i^r$, $k \in [2, r]$ and h_i^k is some integer. The original trapdoor corresponds to $b_i^k = a_i^k$, $U^k = (w^k)^{-1} \bmod p^k$, and $V^k = p^k$, for $k = 1$ to r. To make these equations linear, V^k is set to an arbitrary constant. Consequently, one is not likely to find the original trapdoor but alternate trapdoors can return an alternate superincreasing series according to the following Lemma due to Desmedt, Vanderwalle, and Govaerts [20] and independently Eier and Lagger [18].

Lemma 1. Under assumptions: $b_i^r = a_i^r$ and $b_i^{r-1} \equiv U^r b_i^r \bmod V^r$, then $b_i^{r-1} = \frac{V^r}{p^r}(a_i^{r-1} + \gamma p^r a_i^r \bmod p^r)$, where U^r and V^r are positive integers, $\gamma = \frac{U^r}{V^r} - \frac{\overline{w}^r}{p^r}$ and $\overline{w}^r \equiv (w^r)^{-1} \bmod p^r$.

Proof. Let $b_i^{r-1} \equiv U^r b_i^r \bmod V^r$. Then $b_i^{r-1} = U^r a_i^r - \left\lfloor \frac{U^r a_i^r}{V^r} \right\rfloor V^r = V^r \left(\left(\frac{\overline{w}^r}{p^r} + \gamma \right) a_i^r - \left\lfloor \left(\frac{\overline{w}^r}{p^r} + \gamma \right) a_i^r \right\rfloor \right) = \frac{V^r}{p^r} p^r \left(\frac{\overline{w}^r}{p^r} \left(a_i^r + \frac{p^r}{\overline{w}^r} \gamma a_i^r \right) - \left\lfloor \frac{\overline{w}^r}{p^r} \left(a_i^r + \frac{p^r}{\overline{w}^r} \gamma a_i^r \right) \right\rfloor \right) = \frac{V^r}{p^r}(a_i^{r-1} + \gamma p^r a_i^r \bmod p^r)$, for $i = 1$ to n. ❑

The first stage of Brickell's attack is to find the h_i^k's of (8) by finding short vectors in a lattice containing the public weights a_i^r, for $i = 1$ to n. Numerous other knapsack cryptosystems and cryptanalytical attacks are reviewed in [2]–[5].

3 The Residue Knapsack Public-Key Cryptosystem

3.1 A Dense Multiple-Iterated Knapsack

Merkle and Hellman [6] suggested raising the knapsack density to permit faster signature generation by selecting a dense super-increasing series and using multiple-iterations to randomize the knapsack weights. The density of a *single-iterated* (i.e. $r = 1$) MH knapsack can be close to 1.0 if the initial super-increasing series is very dense such as {1, 2, 4, 8, 17, 35, 68, 142} with $s = 1$. The smallest weights of a dense superincreasing series are always close to a binary progression but this information does not seem to help present attacks on multiple-iterated knapsacks [22]–[24]. The density of a MH knapsack has an upper bound of $\frac{ns}{\log_2 p^r} \approx ns/(ns + (r-1)(s + \log_2 n))$ because of the small-sum principle (5). A problem to be solved is: how can the MH trapdoor be multiple-iterated without lowering the density?

To permit the density to be estimated, we will precisely define the first round of a residue knapsack. A dense superincreasing series is selected with $a_1^0 = 1$ and random weights: $a_i^0 \in (v_i, (1.0 + \mu)v_i)$, for $i = 2$ to n, where $v_i = (2^s - 1)\sum_{j=1}^{i-1} a_j^0$ and $\mu > 0.0$. Next, p^1 is randomly selected from $[(1.0 + \mu/2)v_{n+1}, (1.0 + \mu)v_{n+1}]$; the lower limit ensures that $p^1 - v_{n+1}$ is large to neutralize Shamir's attack for very dense knapsacks [16]. The maximum value of p^1 can easily be shown to be close to $2^{n(s+\mu)}$ if $\mu \in [0.0, 1.0)$ because $(1.0 + \mu) \approx 2^{\mu}$. The density after the first round is greater than $ns/\log p^1 \approx s/(s + \mu)$. If $s = 100$ and $\mu = 2^{-3}$, then the first round density is approx. 0.998 but a_i^0, for $i = 2$ to n, are selected from at least 2^{97} possibilities.

Desmedt, Vandewalle, and Govaerts [20] proposed introducing random *positive* weights at intermediate rounds to reduce the number of "useless" knapsacks. To address the decreasing density of the MH knapsack with multiple-iterations, a *negative* weight $a_{n+k}^k = -p^k$ is introduced after the kth modular multiplication, for $k = 1$ to $r - 1$. During encoding, the variable corresponding to a_{n+k}^k is calculated from the message:

$$x_{n+k} = \left\lfloor \sum_{i=1}^{n+k-1} x_i f_i^k \right\rfloor, \tag{9}$$

where

$$f_i^k = \frac{a_i^k}{p^k}. \tag{10}$$

Desmedt, Vandewalle, and Govaerts [20] proved that satisfying the small-sum principle guarantees unique decoding. The small-sum principle is satisfied if

$$y'^k \triangleq \sum_{i=1}^{n+k} x_i a_i^k \in [0, p^{k+1}), \tag{11}$$

for $k = 1$ to $r - 1$, where $y'^k \equiv y'^{k-1} w^k \bmod p^k$. Adding $x_{n+k} a_{n+k}^k$ reduces $y'^k \bmod p^k$ because $\sum_{i=1}^{n+k} x_i a_i^k \equiv \sum_{i=1}^{n+k-1} x_i a_i^k - \left\lfloor \sum_{i=1}^{n+k-1} x_i \frac{a_i^k}{p^k} \right\rfloor p^k \equiv \sum_{i=1}^{n+k-1} x_i a_i^k \bmod p^k$. The reduction of $y'^k \bmod p^k$ may not be complete because the fractions $f_i^k = a_i^k/p^k \in [0.0, 1.0)$ have finite precision. The fraction precision determines the minimum size of p^{k+1} as shown next.

Proposition 1. Let $p^{k+1} > (1 + \varepsilon)p^k$, fractions f_i^k, for $i = 1$ to $n + k - 1$, are truncated at $s + \log_2(n/\varepsilon) + k - 1$ bits, ε is a positive real, and $x_i \in [0, 2^s)$. Then $y^k \in [0, p^{k+1})$.

Proof. Truncating $f_i^k = \frac{a_i^k}{p^k}$ at $s + \log_2 n/\varepsilon + k - 1$ bits fractional precision results in a truncation error per fraction of $(-\varepsilon 2^{-s-k+1}/n, 0]$. Then the approximation error of x_{n+k} in (9) *before truncation* is $\sum_{i=1}^{n+k-1} x_i f_i^k - \frac{\sum_{i=1}^{n+k-1} x_i a_i^k}{p^k} \in (-\varepsilon, 0]$ because $x_i \in [0, 2^s)$, for $i = 1$ to n, and $x_i \in [0, 2^{s+\log n+i-n-1})$, for $i = n + 1$ to $n + k - 1$. After truncation, $x_{n+k} - \frac{\sum_{i=1}^{n+k-1} x_i a_i^k}{p^k} \in (-1 - \varepsilon, 0]$. Then $y^k \triangleq \sum_{i=1}^{n+k} x_i a_i^k = \sum_{i=1}^{n+k-1} x_i a_i^k - x_{n+k} p^k \in [0, (1 + \varepsilon)p^k) \in [0, p^{k+1})$. ❑

If p^{k+1} is randomly selected from $(p^k(1 + \varepsilon), p^k(1 + \varepsilon)(1 + \mu))$ to satisfy Proposition 1, then the maximum p^k is approximated by

$$p^k < 2^{n(s+\mu)+(k-1)(\varepsilon+\mu)}. \tag{12}$$

The density after the rth round is close to

$$d \approx \frac{ns + (r-1)(s + \log n)}{\log p^r} \tag{13}$$

because there are n variables of s bits and $r - 1$ variables of $s + \log n$ bits. With $r = 2$, $s = 100$, $n = 6$, and $\mu = \varepsilon = 2^{-3}$, the final density is close to 1.17.

3.2 A New Trapdoor Disguising Operation

We have found a new disguising operation that can be appended to any knapsack trapdoor. Consider reducing the knapsack weights a_i^r, for $i = 1$ to $n + r - 1$, by two moduli $\{q_1, q_2\}$:

$$a_{ij} \equiv a_i^r \bmod q_j\ , \tag{14}$$

for $j = 1$ to 2, where q_2 is kept secret, $p^r = q_1 q_2$, and $\gcd\{q_1, q_2\} = 1$. Then $\{a_{i1}, a_{i2}\}$, for $i = 1$ to $n + r - 1$, and q_1 are published. To simplify reduction, q_1 can be a power of two. Recombining the published residues $\{a_{i1}, a_{i2}\}$ to return a_i^r with q_2 secret appears to be challenging as discussed in §4.1.

3.3 Encryption

Encryption is performed with a knapsack group:

$$y_j \equiv \sum_{i=1}^{n+r-1} x_i a_{ij} \bmod q_j, \tag{15}$$

for $j = 1$ to 2, where $r > 1$. Reduction of $y_2 \bmod q_2$ is delayed until decryption because q_2 is secret. The ciphertext is represented by $\{y_1, y_2\}$. The public-key consists of positive integers a_{ij}, for $i = 1$ to $n + r - 1$ and $j = 1$ to 2, fractions f_i^k, for $i = 1$ to $n + k - 1$ and $k = 1$ to $r - 1$, and q_1. The secret-key includes $\{a_1^0, a_2^0, \ldots, a_n^0\}$, p^k and w^k, for $k = 1$ to r, and q_2. Decoding starts by recombining $y^r \equiv \{y_1, y_2\} \bmod \{q_1, q_2\}$ with the Chinese remainder theorem, where

$$y^r = q_2((y_1 - y_2)q_2^{-1} \bmod q_1) + y_2 \tag{16}$$

[8, pp. 268–275]. As with the MH trapdoor, decoding proceeds from y^r according to (6) and (7).

Message encryption is now demonstrated with a small example. First, a private key is selected:

Let $r = 2$, $n = 2$, and $s = 5$
Let $p^1 = 1221$ and $\{q_1, q_2\} = \{256, 9\}$
Then $p^2 = q_1 q_2 = 2304$
Let $w^1 = 845$ and $w^2 = 329$
Let $a_i^0 = \{1, 32\}$, for $i = 1$ to n

Calculating the public key:

$a_i^1 \equiv w^1 a_i^0 \bmod p^1 \equiv \{845, 178\}$, for $i = 1$ to n
$f_i^1 = \dfrac{a_i^1}{p^1} = \{0.69, 0.14\}$, for $i = 1$ to n
$a_{n+1}^1 = -p^1$
$a_i^2 \equiv w^2 a_i^1 \bmod p^2 \equiv \{1525, 962, 1491\}$, for $i = 1$ to $n + 1$
$a_{i1} \equiv a_i^2 \bmod q_1 \equiv \{245, 194, 211\}$, for $i = 1$ to $n + 1$
$a_{i2} \equiv a_i^2 \bmod q_2 = \{4, 8, 6\}$, for $i = 1$ to $n + 1$

Encrypting a message of $\{x_1, x_2\} = \{22, 6\}$:

$$x_{n+1} = \left\lfloor \sum_{i=1}^{n} x_i f_i^1 \right\rfloor = \lfloor 22 \cdot 0.69 + 6 \cdot 0.14 \rfloor = 16$$

$$y_1 \equiv \sum_{i=1}^{n+1} x_i a_{i1} \bmod q_1 \equiv 22 \cdot 245 + 6 \cdot 194 + 16 \cdot 211 \bmod 256 \equiv 202$$

$$y_2 = \sum_{i=1}^{n+1} x_i a_{i2} = 22 \cdot 4 + 6 \cdot 8 + 16 \cdot 6 = 232$$

Decrypting the ciphertext $\{y_1, y_2\} = \{202, 232\}$:

$y^2 \equiv \{202, 232\} \bmod \{256, 9\} \equiv \{202, 7\}$
$y^2 = q_2((y_1 - y_2)q_2^{-1} \bmod q_1) + y_2 = 9((202 - 7)57 \bmod 256) + 7 = 970$
$y^1 \equiv (w^2)^{-1} y^2 \bmod p^2 \equiv 2297 \cdot 970 \bmod 2304 \equiv 122$
$y^0 \equiv (w^1)^{-1} y^1 \bmod p^1 \equiv 302 \cdot 122 \bmod 1221 \equiv 214$
$x'_2 = \lfloor 214/32 \rfloor = 6$
$x'_1 = 214 - 6 \cdot 32 = 22$

The deciphered message $\{x'_1, x'_2\} = \{22, 6\}$ matches the original message.

A further security precaution is to concatenate a message with standard or random bits or sign the message to neutralize a chosen ciphertext attack as described in §4.4. Another security measure is to set $\gcd(a_i^0, p^1) = 1$ to address an attack of §4.2.

3.4 Signatures

Signature generation is similar to MH's method [6] except that the message or hash value is assigned to $y_1 \in [0, q_1)$ and secret random numbers may be assigned to y_2 to neutralize a chosen ciphertext attack described in §4.4 (a fixed integer, secret or public, may also be used). Decoding $\{y_1, y_2\}$ yields the signature $\{x_1, x_2, \ldots, x_n\}$.

Signature generation is repeated with a perturbation of y_2 until $x_i \in [0, 2^s)$, for $i = 1$ to n (to ensure that information about $\{a_1^0, a_2^0, \ldots, a_n^0\}$ and p^1 is not revealed). The average number of signature generation attempts can be shown to be $p^1/2^{ns} \approx 2^{n\mu}$, where μ is defined in §3.1 (a proof of this result is in [47, pp. 168]). There are less than two trials on average when $\mu < 1/n$.

A residue knapsack signature is valid if

$$y_1 \equiv y'_1 - \sum_{k=1}^{r-1} l^k a^r_{n+k,1} \bmod q_1, \tag{17}$$

where y'_1 is the encoded signature and l^k is some integer in the range $[-\lfloor 1 + \varepsilon \rfloor, \lfloor p^{k+1}/p^k \rfloor]$. With corrections by multiples of a^r_{n+k}, the number of signature generation trials depends only on the density of the initial knapsack. A high final density is still minimizes the verification time.

Multiples of a^r_{n+k} account for differences in the completeness of the r modular reductions between the encoder and decoder. A decoder calculates $y^k \in [0, p^{k+1})$, for $k = r - 1$ to 0, as defined by (6) during signature generation. An encoder computes $y'^k \in [0, p^{k+1})$, for $k = 0$ to $r - 1$, following (11) during signature verification. Information is lost during signature generation when y^{k+1} is reduced mod p^k in (6), where $p^{k+1} > (1 + \varepsilon)p^k$ by Proposition 1. Path differences may occur, where $y^k = y'^k + l^k p^k$ and l^k is some integer. The smallest l^k value is $-\lfloor (y'^k_{\max} - y^k_{\min})/p^k \rfloor = -\lfloor ((1 + \varepsilon)p^k - 0)/p^k \rfloor = -\lfloor 1 + \varepsilon \rfloor$ and the largest l^k is $\lfloor (y^k_{\max} - y^k_{\min})/p^k \rfloor = \lfloor (p^{k+1} - 0)/p^k \rfloor = \lfloor p^{k+1}/p^k \rfloor$, where $y^k_{\min} \le y^k \le y^k_{\max}$.

Normally, $p^{k+1}/p^k \approx 1 + \varepsilon < 2$ and the verifier only has to check if $l^k \in \{-1, 0, 1\}$, for $k = 1$ to $r - 1$. The probability that $l^k = 0$ is $p^k/p^{k+1} \approx 1/(1 + \varepsilon)$ and increases with the density. Deviations by multiples of a^r_{n+k} do not significantly delay signature verification if ε or r is small.

The signature generator can ensure that valid signatures have $l^{r-1} = 0$ by combining the message (or hash) value y_1 with a secret random integer c according to the relation

$$y^{r-1} = cq_1 + \left((w^r)^{-1} y_1 \bmod q_1\right), \tag{18}$$

where $c \in [p^{r-1}\varepsilon/q_1, p^{r-1}/q_1)$. Signature generation is completed as usual from y^{r-1}.

If only signatures or identification is required, not encryption or key-distribution, then the knapsack weights only need to be published modulo q_1 instead of modulo $\{q_1, q_2\}$. Also, the condition $p^{k+1} > p^k(1 + \varepsilon)$ of Proposition 1 is not required. In this "signature-only" mode, valid signatures will always have $l^k = 0$, for $k = 1$ to $r - 1$, if *decreasing* moduli $p^k > p^{k+1}(1 + \varepsilon)$

are selected and the fractions f_i^k, for $i = 1$ to $n + k - 1$ and $k = 1$ to $r - 1$, are *rounded up* (instead of truncating) at $s + \log_2 n/\varepsilon + k - 1$ bits precision.

The redundancy, $ns - \log q_1$ bits, of the signature may be reduced by selecting $p^{z+1} \gg p^z$ and $p^{k+1} > (1 + \varepsilon)p^k$ in a signature / encryption mode ($p^k > (1 + \varepsilon)p^{k+1}$ in a signature-only mode), for $k = 1$ to $z - 1$ and $k = z + 1$ to $r - 1$, where $z \in [1, r - 2]$. Setting $z < r - 1$ ensures that $p^{r-1} \gg q_1$ to counter the trapdoor attacks in Section 4.2. Then q_1 can be larger relative to ns because p^{r-1} has increased. In a signature only-mode, a signature is valid if $y_1 \equiv y'_1 - I^z a^r_{n+z,1} \bmod q_1$, where I^z is congruent modulo q_1 to a value in the range $[-\lfloor 1 + \varepsilon \rfloor, \lfloor p^{z+1}/p^z \rfloor]$ and I^z can be calculated according to the relation $I^z \equiv (a^r_{n+z,1})^{-1}(y'_1 - y_1) \bmod q_1$. In a signature / encryption mode, the verifier may also have to recalculate I^z with $I^k = \pm 1$, $k \neq z$. Redundancy is shifted from the signature to y_1 at the expense of a smaller density. Whether a smaller redundancy offsets the lower density depends on the success of cryptanalysis.

Minimum signature size, ns bits, depends on the difficulty of the knapsack problem generated (see §4.3). The smallest signature is with $n = 2$, $s \geq 130$, and $r \geq 4$ or $n = 200$ and $s = 1$. A signature forger may attempt to sign a message with a linear combination of previous signatures. Concatenating standard data with the message or using a secure hash function will neutralize combination attacks. A hash function of 128 bits is recommended to counter birthday attacks [46] and this requires $\log q_1 > 128$.

3.5 Generating Common Knapsack Weights

The public-key size can be reduced by letting r of the total $n + r - 1$ knapsack weights be common for all parties in a network. Encoding and decoding are not changed. There are still the same number of possible private-keys for a given public-key. The number of possible public-keys is reduced but is still very large. Present cryptanalytic attacks do not appear to be stronger with r common weights.

The knapsack weights $a^r_{ij} \triangleq a_{ij}$, for $i = n$ to $n + r - 1$ and $j = 1$ to 2 may be common, as well as q_1. Common values of $a_{i,1}$ are arbitrarily selected from the range $[q_1/2, q_1)$. A lower limit $q_1/2$ avoids small weights that weaken the knapsack problem. Common values of $a_{i,2}$ are selected from $[q'_2/2, q'_2)$, where the secret q_2 has a minimum or average value of q'_2.

Then the private-key values: $\{a_1^0, a_2^0, \ldots, a_n^0\}$, $\{p^1, p^2, \ldots, p^r\}$, and q_2, are secretly selected as usual. Next, the common weights $\{a^r_{i1}, a^r_{i2}\}$ are recombined with the Chinese remainder theorem [8, pp. 268–275], where $a_i^r \equiv \{a^r_{i1}, a^r_{i2}\} \bmod \{q_1, q_2\}$, for $i = n$ to $n + r - 1$. Then $\{w^1, w^2, \ldots, w^r\}$ are calculated according to the relation

$$w^k \equiv (a^{k-1}_{n+k-1})^{-1} a^k_{n+k-1} \bmod p^k, \tag{19}$$

decrementing sequentially from $k = r$ to 1, where $a^{k-1}_{n+k-1} = -p^{k-1}$ and a^k_{n+k-1} is calculated according to the relation $a^{h-1}_{n+k-1} \equiv (w^h)^{-1} a^h_{n+k-1} \bmod p^h$, decrementing sequentially from $h = r$ to $k + 1$. The private-key is now fully defined and the non-common parts of the public-key may be calculated as usual. When r weights and q_1 are common, the public-key size is

$$(n - 1)\log p^r + ((r - 1)n + \sum_{i=1}^{r-2} i)(s + \log n/\varepsilon + r - 2) \tag{20}$$

bits ($\log p^r$ approaches ns as μ and ε approach zero according to (12)).

Normal key-generation with the private-key including the calculated $\{w^1, w^2, \ldots, w^r\}$ does not always return the common weights. When calculating a^k_{n+k-1} to find w^k, if $a^h_{n+k-1} \geq p^h$, where $h \in [k, r]$, then a^r_{n+k-1} will not match its common value. The probability p^h/p^{h+1} that $a^h_{n+k-1} < p^h$ increases with the density. If this test is not passed, a small permutation of the common weight a^r_{n+k-1} may be tried. For example, increment $a^r_{n+k-1} \bmod q_1$ by one and recalculate a^k_{n+k-1}. The small difference between the actual a^r_{n+k-1} and the common a^r_{n+k-1} is published as part of the public-key. Occasionally, a value of w^k is found that is not relatively prime to p^k and then the inverse of $w^k \bmod p^k$ does not exist. Again, a^r_{n+k-1} may be slightly modified and w^k recalculated. Also, $w^k = 0$ will occur if $a^r_{n+k-1} = a^r_{n+k}$. Any difference between the common weights ensures that $w^k \neq 0$. An area for future work is to generate some common fractions f^k_i in the public-key plus the above common weights.

4 Cryptanalysis of the Residue Knapsack

4.1 On the Residue Recombination Problem

Reducing a^r_i modulo $\{q_1, q_2\}$ with q_2 kept secret as described in §3.2 creates a residue recombination problem. If q_2 were publicly known, an attacker could easily recombine $a^r_i \equiv \{a_{i1}, a_{i2}\} \bmod \{q_1, q_2\}$ by the Chinese remainder theorem [8, pp. 268–275]. In §4.2, we will describe how to unwind the high-density modular multiplications of the residue knapsack trapdoor given a^r_i. Exhaustively searching for q_2 and unwinding the trapdoor for each guess is not feasible with $q_2 > 2^{\partial/2}$ assuming each trapdoor trial solution requires $2^{\partial/2}$ operations and $\partial \approx 80$.

Brickell's attack [24] as reviewed in §2.2 finds an alternate trapdoor, where $Vr \neq p^r$. Finding q_2 does not appear to be easier than finding p^r because $p^r = q_1 q_2$. What are the consequences to Brickell's attack of an alternate recombination? Suppose an attacker uses an arbitrary modulus m_2 instead of q_2 to recombine $\{a_{i1}, a_{i2}\}$, where $Vr = q_1 m_2$. Recombining the residues of the knapsack weights with $m_2 \neq q_2$ linearly shifts a^r_i by a multiple of q_1 that varies with i.

Lemma 2. Under the assumption that $a^r_i \equiv \{a_{i1}, a_{i2}\} \bmod \{q_1, q_2\}$, $b^r_i \equiv \{a_{i1}, a_{i2}\} \bmod \{q_1, m_2\}$, $q_2 \in [c, c+d)$, $m_2 \in [c, c+d)$, and c and d are positive integers, then $(a^r_i - b^r_i)/q_1$ is some integer in the range $(-c-d, c+d)$, for $i = 1$ to $n + r - 1$.

Proof. If $a^r_i \equiv \{a_{i1}, a_{i2}\} \bmod \{q_1, q_2\}$, then $a^r_i = a_{i1} + q_1(a_{i2} - a_{i1})q_1^{-1} \bmod q_2$ by the Chinese remainder theorem. Similarly, $b^r_i = a_{i1} + q_1(a_{i2} - a_{i1})q_1^{-1} \bmod m_2$. Then $(a^r_i - b^r_i)/q_1 = ((a_{i2} - a_{i1})q_1^{-1} \bmod q_2) - ((a_{i2} - a_{i1})q_1^{-1} \bmod m_2) \in (-c-d, c+d)$. ❑

Random and superincreasing images $\{b_1^{r-1}, b_2^{r-1}, \ldots, b_n^{r-1}\}$ are found with small-sum modular mappings (SSMMs) that satisfy (8). These SSMMs create similar knapsack problems (i.e. different knapsack problems with the same solution). With n similar knapsacks, a set of linear equations can be solved to find the message. Shamir's compact knapsack [44] attack employs enumeration to find SSMMs and Brickell's low-density attack uses lattice basis reduction [30]. Random images exist for multiple-iterated MH knapsacks when γ approaches zero by Lemma 1.

In a residue knapsack, $\{b_1^{r-1}, b_2^{r-1}, \ldots, b_n^{r-1}\}$ has a negligible probability of being a random image when γ approaches zero because of the linear shifts. Lemma 1 does not guarantee the existence of random or superincreasing images to a residue knapsack. Numerical experiments confirmed that $\gamma \to 0$ is not sufficient to return a superincreasing series when $m_2 \neq q_2$.

The linear shifts caused by an alternate recombination are similar to those of the "linear shift knapsack" of Laih et. al [15]. The probability of random and superincreasing images existing for a random knapsack is $nV^r/n!$ [15] and $2^{-\binom{n}{2}} \cdot \left(\sum_{i=1}^{n} a_i\right)^2$ [19] respectively with $s = 1$. It still remains to be proven whether the linear shifts are sufficiently random to ensure that alternate superincreasing images of a residue knapsack have a small probability of existing.

Signatures are checked modulo the public q_1 and in a signature-only mode, a_{i2} does not need to be published. Then a smaller $q_2 > 2^{\partial/2(n+r-1)}$, where $\partial \approx 80$, foils an exhaustive search for a_{i2}, for $i = 1$ to $n + r - 1$. The reader may wonder if a superincreasing series could be recovered from a_{i1} and q_1? Lemma 1 implies that the alternate weights $b_i^{r-1} \equiv U^r a_{i1} \bmod V^r$ approach $V^r(a_i^{r-1} \bmod q_1)/q_1$ as γ' approaches 0, where $\gamma' = U^r/V^r - ((w^r)^{-1} \bmod q_1)/q_1$. In a multiple-iterated knapsack, a_i^{r-1} has an *average* value of $p^{r-1}/2$ and b_i^{r-1} approaches $V^r(a_i^{r-1} - t_i q_1)/q_1$ as γ' approaches 0, where t_i is some integer in $[0, p^{r-1}/q_1)$. Then unwinding $a_{i1} \bmod q_1$ linear shifts b_i^{r-1} by a multiple of q_1 as large as p^{r-1}/q_1.

4.2 Unwinding the Residue Knapsack Trapdoor

Brickell's multiple-iterated knapsack attack [24] depends on the information leaked by the MH integer knapsack weights as expressed by (8) but the residue knapsack follows only (11). Brickell's experimental evidence [24] shows that his multiple-iterated knapsack attack is not successful unless $p^{k+1}/a_i^k > 2^3$, for $i = 1$ to n and $k = 1$ to $r - 1$ (with $s = 1$ and $n \approx 50$). MH's knapsack has $p^{k+1}/a_i^k \approx 2^{s+\log n}$ by the small-sum principle (5), which is always sufficient for Brickell's attack. A residue knapsack has $p^{k+1}/a_i^k \approx 1 + \varepsilon \approx 1.125$ assuming $\varepsilon = 2^{-3}$, which is never sufficient for Brickell's attack.

Shamir [16] showed that modular multiplications can become a vulnerable permutation as p^{k+1}/p^k approaches 1.0. Shamir's attack has time $O(n^{3/2}v^{1/2})$, where $v = p^{k+1} - p^k$. With a residue trapdoor, $v > \varepsilon p^k > \varepsilon 2^{ns}$. For example, with $ns = 200$ and $\varepsilon = 2^{-3}$, Shamir's attack has over 2^{98} operations. The residue knapsack has a p^{k+1}/p^k ratio well below the range of Brickell's [24] attack but safely beyond the range of Shamir's [16] attack.

If the integer weights of a residue knapsack do not leak sufficient information for a simultaneous diophantine approximation attack, then we can still employ the fractions $f_i^k = a_i^k/p^k$, for $i = 1$ to $n + k - 1$ and $k = 1$ to $r - 1$. The numerators a_i^k and constant denominator p^k, where $a_i^k \in [0, p^k)$ and $\log p^k > ns$, could be recovered by Stern and Toffin's [25] lattice basis reduction attack if the fractions were published to sufficient precision. A minimum fraction precision to recover a set of numerators and a common denominator is generally determined next.

Lemma 3. Under the assumption of integers $a_i \in [0, p)$, $p \in [0, 2^r)$, and fractions $f_i \in [0.0, 1.0)$, for $i = 1$ to n, then the average number of sets of $\{a_1, a_2, \ldots, a_n, p\}$ such that $\left|f_i - \frac{a_i}{p}\right| < 2^{-L}$ for a given set of $\{f_1, f_2, \ldots, f_n\}$ is greater than $2^{(r-1)(n+1)-nL}$.

Proof. Under the further restrictions: $p \in [2^{t-1}, 2^t)$ and $a_i \in [0, 2^{t-1})$, there are $2^{(t-1)(n+1)}$ possible sets of $\{a_1, a_2, \ldots, a_n, p\}$. In practice, the fractions $f_i = a_i/p$, for $i = 1$ to n, are evenly distributed over [0.0, 1.0). If each fraction is precise to L bits, then the total number of possible sets of fractions $\{f_1, f_2, \ldots, f_n\}$ is 2^{nL}. Then a given set $\{f_1, f_2, \ldots, f_n\}$ will satisfy $\left|f_i - \frac{a_i}{p}\right| < 2^{-L}$ for $2^{(t-1)(n+1)-nL}$ sets of $\{a_1, a_2, \ldots, a_n, p\}$ on average. ❑

Lemma 3 shows that the original numerators and common denominator can be isolated from the fractions on average if $(t-1)(n+1) - nL < 0$ or, equivalently, $L > (t-1)(1 + 1/n) \approx t(1 + 1/n)$. Stern and Toffin's [25] numerical experiments found a minimum fraction precision a little larger than $t(1 + 1/n)$ bits. As a security measure, if the fraction precision is $t(1 + 1/n) - \partial/n$ bits, then an average of 2^{∂} spurious solutions will satisfy the published fractions. With a residue knapsack, the fraction precision $s + \log n/\varepsilon + k - 1$ bits is always far less than the lower bound $ns(1 + 1/(n + k - 1))$.

By combining the fractions with the integer knapsack weights, larger fractions with the same common denominator may be obtained. Consider an alternate trapdoor for the residue knapsack defined by $b_i^k \equiv W^k b_i^{k-1} \bmod V^k$, for $k = 1$ to r, where the original trapdoor corresponds to $b_i^k = a_i^k$, $W^k = w^k$ and $V^k = p^k$. Each iteration of the alternate trapdoor has to follow the small-sum principle to generate a similar knapsack with the same solution. With a residue knapsack, this requires $V^{k+1} > (1 + \varepsilon)V^k$ and

$$b_i^k - f_i^k V^k < V^k/2^{s+\log n/\varepsilon + k - 1}, \tag{21}$$

for $i = 1$ to $n + k - 1$ and $k = 1$ to $r - 1$. Let

$$z_i^k \equiv \left(-b_{n+k-1}^k\right)^{-1} b_i^k \bmod V^k. \tag{22}$$

Substituting $b_{n+k-1}^k \equiv -W^k V^{k-1} \bmod V^k$ and $b_i^k \equiv W^k b_i^{k-1} \bmod V^k$ in (22), we find

$$b_i^{k-1} \equiv z_i^k V^{k-1} \bmod V^k. \tag{23}$$

Equation 23 can be expressed as $z_i^k V^{k-1} \equiv f_i^{k-1} V^{k-1} \bmod V^k$ by substituting $b_i^{k-1} = f_i^{k-1} V^{k-1}$. Let $z_i^k V^{k-1} = f_i^{k-1} V^{k-1} + h_i^k V^k$, where h_i^k is some integer in the range $[0, V^{k-1})$. This equation is divided by V^k and V^{k-1} to generate the fractions

$$F_i^{k-1} \triangleq \frac{z_i^k - f_i^{k-1}}{V^k} = \frac{h_i^k}{V^{k-1}}, \tag{24}$$

for $i = 1$ to $n + k - 1$. The fractions F_i^{k-1} have a common denominator V^{k-1}, numerators $h_i^k \in [0, V^{k-1})$, and accuracy $\log V^k + s + \log n/\varepsilon + k - 2$ bits.

A residue knapsack trapdoor is unwound one round at a time, from $k = r$ to 2, because the fractions of each round are different. Beginning with $k = r$, this attack attempts to find V^{r-1}. Solving the residue recombination problem such that $V^r = p^r$ and $b_i^r = a_i^r$ appears to be necessary to return a superincreasing series because of the linear shifts of Lemma 2. The attacker knows

$\{a_{i1}, a_{i2}\}$ and q_1, where $a_i^r \equiv \{a_{i1}, a_{i2}\} \bmod \{q_1, q_2\}$, but does not know a_i^r or $p^r = q_1 q_2$ because q_2 is secret.

Suppose that the attacker sets $V^r = q_1$ and $b_i^r = a_{i1}$ and attempts to reconstruct the original common denominator $V^{r-1} = p^{r-1}$. A lower bound on the fraction precision to recover p^{r-1} is $\log p^{r-1}(1 + 1/(n + r - 2))$ bits by Lemma 3. The fractions F_i^{k-1} are accurate to $\log q_1 + s + \log n/\varepsilon + k - 2$ bits with $V^r = q_1$. Then the following security measure is sufficient:

$$\log q_1 < \log p^{r-1} + \frac{\log p^{r-1} - \partial}{n + r - 2} - s - \log n/\varepsilon - r + 2 \tag{25}$$

where $\partial \approx 80$. This can also be expressed as

$$(n + r - 2)(s + \log n/\varepsilon + r - 2 - \log q_2) < \log p^{r-1} - \partial, \tag{26}$$

assuming $\log q_1 + \log q_2 \approx \log p^{r-1}$. Equation 26 corresponds to the following information theoretic argument: the information revealed by the fractions f_i^k, $(n + r - 2)(s + \log n/\varepsilon + r - 2)$ bits, minus the information lost by keeping q_2 secret, $(n + r - 2)\log q_2$ bits, has to be less than a lower bound on the information to unwind one round of the trapdoor, $\log p^{r-1}$ bits, minus a security margin of ∂ bits.

Fractions F_i^1 with a smaller non-common denominator $c_i \triangleq p^1/a_i^0$ can be found when $r = 2$ and a_i^0 divides p^1. Observe that $f_i^1 = a_i^1/p^1 = (w^1 a_i^0 \bmod p^1)/p^1 = (w^1 a_i^0 \bmod c_i a_i^0)/c_i a_i^0 = (w^1 \bmod c_i)/c_i$ and $f_i^1 c_i$ is an integer. Let $z_i^1 \equiv (a_{n+1}^2)^{-1} a_i^2 \equiv (p^1)^{-1} a_i^1 \equiv (p^1)^{-1} f_i^1 p^1 \equiv (c_i a_i^0)^{-1} f_i^1 c_i a_i^0 \equiv c_i^{-1} f_i^1 c_i \bmod q_1$. Fractions $F_i^1 = \dfrac{z_i^1 - f_i^1}{q_1} = \dfrac{h_i}{c_i}$, for $i = 1$ to n, can then be obtained similar to (24). Both c_i and h_i have $\log c_i$ bits and F_i^1 has $\log q_1 + s + \log n/\varepsilon$ bits. Then c_i can be isolated if $\log q_1 + s + \log n/\varepsilon < 2\log c_i$ by Lemma 3. A precaution is to select $\gcd(a_i^0, p^1) = 1$ if $r = 2$ for the largest a_i^0. Other initial knapsack constructions are analyzed in [47, pp. 175–177, pp. 198].

A lattice resembling Brickell's [24] may be generated with a variation on the above attack. Let (23) be expressed as $b_i^{k-1} = z_i^k V^{k-1} - h_i^k V^k$, where h_i^k is some integer in the range $[0, V^{k-1})$. If we replace k with $k - 1$ in (21) and then substitute the above expression for b_i^{k-1}, we find

$$(z_i^k - f_i^{k-1})V^{k-1} - h_i^k V^k < V^{k-1}/2^{s+\log n/\varepsilon + k - 1}, \tag{27}$$

for $i = 1$ to $n + k - 2$. Then

$$| g_i^k h_1^k - g_1^k h_i^k | \le d^k p^{k-1}, \tag{28}$$

for $i = 2$ to $n + k - 1$, where $k \in [2, r]$, $g_i^k = (z_i^k - f_i^{k-1})d^k$, and $d^k = 2^{s+\log n/\varepsilon + k - 1}$ (the factor d^k converts f_i^{k-1} to an integer). The h_i^k's are found by reducing the lattice [26]–[29] of (29). One round is unwound at a time because the fractions of each round are different. If the attacker sets $V^r = q_1$ and $b_i^r = a_{i1}$, then the same counter-measure is effective against this attack: $q_1 < \max(q_1)$, where $\max(q_1)$ is defined by (25) with q_2 secret.

$$\begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ g_2^k & g_1^k & 0 & \cdots & 0 \\ g_3^k & 0 & g_1^k & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_n^k & 0 & 0 & \cdots & g_1^k \end{pmatrix} \tag{29}$$

An adaption of Adleman's trapdoor attack [23] to the residue knapsack is described in Appendix A. That attack is not feasible with $q_1 < \max(q_1)$. To ensure that these adaptions of the trapdoor attacks of Stern–Toffin [25], Brickell [24], and Adleman [23], are all not feasible, we select a secret q_2 of magnitude specified by (26). Any of these attacks appear to be capable of unwinding the high-density modular multiplications of the residue trapdoor but not of solving the residue recombination problem.

We considered using the counter-measure of selecting $\gcd(a^r_{n+r-1}, p^r) = c$ for the above attacks, where c is sufficiently large, with the key selection technique of §3.5 because then the inverse of b^r_{n+r-1} modulo V^r will not exist when calculating z^r_i with $V^r = p^r$ and $b^r_{n+r-1} = a^r_{n+r-1}$. This seems to require either $\gcd(w^r, p^r) = c$, obstructing decoding, or $\gcd(p^{r-1}, p^r) = c$, and then the final round may only need to be unwound modulo p^r/c. In any case, setting $\gcd(a^r_{n+r-1}, p^r) \neq 1$ does not neutralize Adleman's attack. Consequently, a secret q_2 is essential to the trapdoor security.

4.3 On the Residue Knapsack Problem

The residue knapsack problem: (15) plus (9), can not be easier than the contained NP-complete [10] classical knapsack group with n variables. The question to be answered is whether the residue knapsack problem is any harder than the contained knapsack group? Are known solutions to the knapsack problem adaptable to (9)? At low density, (15) will have a single solution (the original message) and (9) can be ignored. Equation 15 becomes underdetermined without (9) as the density is increased.

Consider a classical knapsack with ns message bits, largest weight A, $\log A + s + \log n$ ciphertext bits, and density $d \approx ns/\log A$. Classical knapsacks can easily be shown to have $2^{-e-s-\log n}$ solutions on average, where $e \triangleq \log A - ns$. The average number of solutions can also be expressed as $\dfrac{(A)^{d-1}}{2^{s+\log n}}$ and grows exponentially with the density. A classical knapsack has 2^c solutions when the density is $d\{2^c\} = 1 + \dfrac{s + \log n + c}{\log A}$. The virtual d_{max} to permit unique decoding corresponds to $d\{2^c\}$ at $c = 0$.

Schnorr and Euchner [38] found empirically that low-density attacks of the Lagarias and Odlyzko [32] type have the lowest probability of success when $d \approx 1 + (\log(n/2))/n$. This is very close to the virtual d_{max}. Apparently, finding solutions gets easier beyond the virtual d_{max} but the message becomes obscured by spurious solutions. If we set $c = 40$, then there are 2^{40} spurious solutions and exhaustively computing 2^{40} lattice basis reductions is not feasible.

With a residue knapsack, there are $ns + (r - 1)(s + \log n)$ total variable bits, the largest weights is $A \approx p^r$, $\log p^r + s + \log n$ ciphertext bits, and density $d = \dfrac{ns + (r-1)(s + \log n)}{\log p^r}$. The average number of solutions is $2^{(r-2)(s+\log n)-e}$, where $e = \log p^r - ns$, or $\dfrac{(p^r)^{d-1}}{2^{s+\log n}}$.

A residue knapsack with 2^c solutions has a density $d\{2^c\} = 1 + \frac{s + \log n + c}{\log p^r}$. According to (12), $\log p^r \approx n(s + \mu) + (r - 1)(\varepsilon + \mu)$ and $e \approx n\mu + (r - 1)(\varepsilon + \mu)$. In a compact mode, (15) independently has 2^{40} solutions if $r \geq 3$ and $s > 40$ assuming $e = 1$. In a [0, 1] knapsack, (15) has 2^{40} solutions if $r \geq 8$ and $n > 200$ assuming $e = \log n$.

The residue knapsack problem for signatures: (15) with $j = 1$, has similar density levels. A signature has ns bits and the message-to-be-signed has $\log q_1$ bits. There are $ns - \log q_1$ redundant bits in the signature that can be preset to an arbitrary value to lower the density (presetting was first used by Odlyzko [33]). Presetting reduces the total number of variable bits to $\log q_1 + (r - 1)(s + \log n - e'/n)$, where $e' \triangleq ns - \log q_1$. A signature residue knapsack problem has a density $d' = 1 + \frac{(r - 1)(s + \log n - e'/n)}{\log q_1}$ after presetting. The average number of solutions after presetting is $2^{(r-1)(s+\log n - e'/n)}$ or, equivalently, $(q_1)^{d'-1}$. Then $d'\{2^c\} = 1 + c/\log q_1$. After presetting, we expect only one valid signature that satisfies (15) plus (9) but there are 2^c spurious solutions on average to (15) independently.

The signature redundancy can be eliminated as described in §3.4. This also reduces the density because q_1 is larger but may foil future attacks exploiting redundancy. Presetting $\log(p^{z+1}/p^z)$ signature bits to further reduce the density is not feasible because knapsack problems with all p^{z+1}/p^z values of I^z have to be solved exhaustively to expect a signature to exist.

Amirazizi, Kamin, and Reyneri's algorithm [43] for the compact knapsack problem involves translating the knapsack problem into an integer programming problem [41],[42]. Applying their algorithm to the residue knapsack problem is not feasible when $c \geq 40$ because translating (9) to an integer programming problem requires multiplications by the fractions f_i, for $i = 1$ to n, that cause large error propagations (the fractions are precise to $s + \log n/\varepsilon$ bits).

Enumeration algorithms [39],[40] do not require multiplications by the fractions and can be applied to the residue knapsack at any c value. Ferreira's algorithm [40] has a time/hardware tradeoff $T \cdot H = O(2^{ns/2})$. We do not know of any algorithms besides enumeration that can be directly applied to a residue knapsack when $c \geq 40$. This difficulty is removed if the residue knapsack can be successfully translated to a similar classical knapsack problem. A similar knapsack problem is defined as a different knapsack problem with the same solution. Then any classical knapsack attack can be applied to the resulting similar classical knapsack problem.

This translation is closely related to the trapdoor attacks of §4.2, as low-density attacks resemble attacks on the MH trapdoor. In the case of the residue knapsack, the intruder looks for small-sum modular mappings (SSMMs) that satisfy (21). These modular mappings define an alternate trapdoor but the resulting initial knapsack just has to be similar, not superincreasing.

The attacker begins the translation by recombining $b_i^r \equiv \{a_{i1}, a_{i2}\}$ modulo $\{q_1, m_2\}$, for $i = 1$ to $n + r - 1$, where $V^r = q_1 m_2$ and $m_2 > \max(y_2)$ is an arbitrary replacement for q_2. Any of the trapdoor attacks of Section 4.2 or Appendix A can be employed to find the SSMMs. The lattice of (29) has the same form as the one used by Brickell's low-density attack [30]. The effective total number of knapsack variables N is $n + r - 2$ when finding SSMMs that satisfy (27) with $k = r$. We considered using the counter-measure $\gcd(a^r_{n+r-1}, q_1) > \partial/2$ to ensure that the inverse of $a^r_{n+r-1} \bmod q_1$ does not exist when finding z_i^r. This is intended to force the adversary to include q_1 as an extra negative knapsack weight to increase N to $n + r - 1$ and reduce the minimum public-key size by about half. With the problems obtaining a large gcd described in §4.2, we estimate N at $n + r - 2$ variables to be safe.

The recombined b_i^r is linear shifted with respect to a_i^r as shown by Lemma 2. Random images (i.e. similar knapsacks) of a random [0, 1] knapsack have a probability of existing less than $nV^r/n!$ as shown by Laih et. al [15]. Then SSMMs would probably not exist at large n unless $V^r \gg p^r$. Recombining with a large V^r artificially lowers the density but fixes the modulus V^r of the SSMM (unless V^r is treated as another negative weight) and linearly shifts the knapsack. Low-density attacks of the Lagarias and Odlyzko type do not require prior recombination but can not be applied directly when $c \geq 40$.

Brickell's low-density attack [30] uses lattice basis reduction to find SSMMs and has a critical density of 0.54. Jorissen et. al [31] showed how to raise this critical density but then the time depends exponentially on s. Finding SSMMs does not get easier beyond the virtual d_{max}.

Integer programming [41],[42] appears to be capable of finding the SSMMs with $N \leq 4$ variables. Shamir's algorithm for a compact knapsack [44] finds SSMMs by enumeration. The complexity of Shamir's algorithm is determined by an enumeration of a [0, 1] knapsack with z variables and $T{\cdot}H = O(2^{z/2})$ using Ferreira's enumeration algorithm [40], where $z = s + N(\log z - 1) \approx s + N6.3$ (assuming $z = 160$) with N knapsack variables. A classical knapsack has $N = n$ variables and a residue knapsack has $N = n + r - 2$ effective total variables.

Ranges of algorithms for the classical knapsack problem with 2^{60} operations are plotted in Fig. 1. A parallel computer with a thousand processors at 10 nanosec/operation can execute 2^{60} operations in half a year.

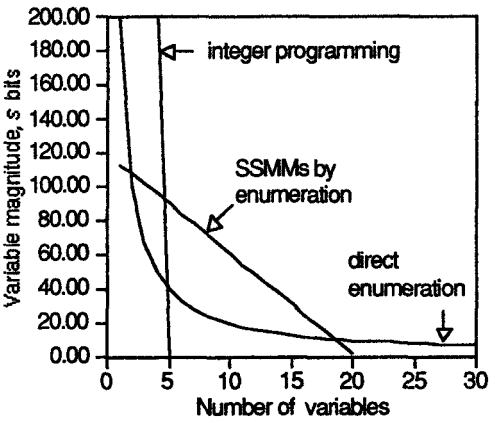

Fig. 1. Feasible ranges of algorithms for the classical knapsack problem with 2^{60} operations

Translating a residue knapsack appears to be close in difficulty to solving a classical knapsack problem with $N = n + r - 2$ variables because there are effectively N variables in the rth round. One SSMM is needed in the rth translation round compared to n SSMMs to solve a classical knapsack. This does not change the estimates of the complexity of Shamir's attack because linear dependencies are rare at small n and n SSMM's for a classical knapsack are found with one enumeration [44]. Then Fig. 1 applies to the residue knapsack when the number of variables corresponds to $N = n + r - 2$.

Exemplary parameters are given in Table 1 including the number of message variables n, variables size s bits, total number of variables $N = n + r - 2$, minimum q_2 to satisfy (26) with $\partial = 80$, public-key (PK) size in kilobits with r common weights as expressed by (20), the density

assuming $\mu = 2^{-5}$ and $\varepsilon = 2^{-3}$ according (12) and (13), and $z/2$, where the strongest known attack on the residue knapsack problem has $T{\cdot}H = O(2^{z/2})$ using Ferreira's enumeration algorithm to find SSMMs in a compact mode and to enumerate directly with [0, 1] knapsacks. For example, if a residue knapsack has $r = 5$, $n = 2$, and $s = 130$, then $N = 5$ and Shamir's algorithm has $T{\cdot}H = O(2^{z/2}) = O(2^{78})$, Kannan's integer programming algorithm [42] has time $O(N^{9N}s) = O(2^{111})$, and direct enumeration has $T{\cdot}H = O(2^{ns/2}) = O(2^{130})$.

Table 1. Exemplary residue knapsack parameters.

r	n	s	ns	N	$\log q_2$	d	$z/2$	PK
5	2	150	300	5	113	3.01	91	2.5
4	3	150	450	5	83	2.01	91	2.8
5	2	170	340	5	125	3.00	101	2.8
5	3	170	510	6	106	2.34	104	4.2
4	4	170	680	6	77	1.76	104	4.7
3	200	1	200	201	12	1.05	100	45.

In general, the difficulty of the residue knapsack problem increases with n, r, and s. The total number of knapsack variables, $N = n + r - 2$, can be increased by raising r instead of n. A small n value permits a smaller signature ns bits, ciphertext $(n + 1)s$ bits, and public-key, and a higher density. The residue knapsack density is not limited by the small-sum principle or the virtual d_{max} but grows with r and s. As the density increases, a larger q_2 is needed to satisfy (26) ($\log_2(q_2) > 40$ is also required except in a signature-only mode as shown in §4.1). Signature redundancy increases with q_2 unless using the redundancy reduction technique of §3.4 that lowers the encryption density. This establishes a limit on encryption density depending on the success of attacks based on redundancy or small q_1.

With a classical knapsack, there are statistical dependencies between the least significant bits of the ciphertext and message variables, especially in a compact mode [7]. This weakness is not present in a residue knapsack because the least significant bits of the extra variables (9) depend on the whole message.

4.4 A Chosen Ciphertext Attack

To find p^1, an adversary doubles both ciphertext residues $\{y_1, y_2\}$ to generate the message-ciphertext pairs (x, y^r) and $(x', 2y^r)$, where $y^r \equiv \{y_1, y_2\} \bmod \{q_1, q_2\}$, and $y'^r = 2y^r \equiv \{2y_1, 2y_2\} \bmod \{q_1, q_2\}$. If $y^0 > p^1/2$, then $p^1 = 2y^0 - y'^0$ with probability near 2^{-r}.

Before finding y^0, the attacker has to find the initial weights $\{a_1^0, a_2^0, \ldots, a_n^0\}$. In a very dense compact knapsack, the attacker knows that $a_1^0 = 1$. The attacker finds the pairs (x, y^r) and $(x', 2y^r)$ starting with a chosen plaintext x, where $x_1 > 2^{s-1}$ and $x_i < 2^{s-2}$, for $i = 2$ to n. Next, the equation $x'_1 + x'_2 a_2^0 = 2x_1 + 2x_2 a_2^0$ is solved for a_2^0. This attack is continued until all the initial weights are known.

Whether this attack can be continued to find the rest of the trapdoor after finding p^1 is not certain but one of several counter-measures can be used. Chosen-ciphertext attacks can be detected if there is some redundancy in the initial knapsack so that all values of y'^0 will not be decodable. As well, the attacker does not know the message corresponding to the chosen-ciphertext and can

not sign the message. Similarly, standard data fields can be used to detect a chosen ciphertext attack. A chosen ciphertext attack can be neutralized by including random data fields that are discarded upon decryption. When generating signatures, y_2 may be set to a random value or any fixed value.

5 Summary

The residue knapsack cryptosystem is potentially the first knapsack to include all of the following features together: a high-density, compactibility, a fast trapdoor, and fast signature generation. The density approaches $1 + (r - 1)(s + \log n)/ns$, well beyond the critical density of low-density attacks [30]–[38]. Encoding has computation $O((ns)^2 + rns(s + \log n/\varepsilon))$, where n is the number of message terms per block of s bits each, r is the number of iterations of modular multiplication in the trapdoor ($r > 1$), and ε is a parameter typically 2^{-3}. Decoding can be performed by Henry's algorithm [45] with computation and memory $O(r(ns)^2)$ and Orton [48] has described a decoding algorithm with computation and memory $O((ns)^2 + rns\log n)$. The public-key size is close to $(n - 1)ns + ((r - 1)n + \sum_{i=1}^{r-2} i)(s + \log n/\varepsilon)$ bits assuming that rns bits of the public-key are common to all parties as proposed in §3.5. Key generation has time $O(rn^3s^2)$. The maximum information rate is $ns/((n + 1)s + \log n)$.

The residue knapsack problem can be solved by enumeration [39],[40] in time-hardware $O(2^{ns/2})$ or by translating the residue knapsack problem to a classical knapsack problem. This translation does not appear to be easier than solving a knapsack problem of $n + r - 2$ variables of s bits each.

Low-density attacks [30]–[38] with lattice basis reduction [26]–[29] have been successfully applied to the [0, 1] knapsack problem but not the compact knapsack problem. Considering present solutions to the knapsack problem, the public-key may be an order of magnitude smaller for a given security level with the compact knapsack problem.

The residue knapsack trapdoor is the first to depend on the simultaneous diophantine approximation problem *plus* the residue recombination problem. Our adaptions of lattice basis reduction attacks [23]–[25] to the residue trapdoor can not feasibly solve the residue recombination problem when the secret q_2 exceeds the minimum magnitude of (26). The reader probably does not need any encouragement to challenge the security for themselves. Several specific parameter sets are suggested in Table 1.

Appendix A: Adleman's Trapdoor Attack

To satisfy the small-sum principle, an alternate residue trapdoor follows the relations:

$$U^k b_i^k - h_i^k V^k - f_i^{k-1} V^{k-1} < V^{k-1}/2^{s+\log n/\varepsilon + k - 1} \tag{30}$$

and

$$U^k b_{n+k-1}^k - h_{n+k-1}^k V^k - V^{k-1} = 0, \tag{31}$$

for $i = 1$ to $n + k - 2$, in the kth round, where $k \in [2, r]$ and h_i^k is some integer. The original trapdoor corresponds to $b_i^k = a_i^k$, $U^k = (w^k)^{-1} \bmod p^k$, and $V^k = p^k$, for $k = 1$ to r. These equations can be fitted to a lattice similar to Adleman's lattice [23]. If the attacker sets $V^r = q_1$ and $b_i^r = a_{i1}$, then the security-measure of (26) is sufficient. The attacker can attempt to recombine $\{a_{i1}, a_{i2}\} \bmod \{q_1, q_2\}$ with the Chinese remainder theorem [8, pp. 268–275], although q_2 is secret, according to the relation:

$$a_i \equiv \sum_{j=1}^{2} a_{ij} Q_j \left(Q_j^{-1} \bmod q_j \right) \bmod p^2 \tag{32}$$

where $Q_j = p^2/q_j$ and $p^2 = q_1 q_2$. Then the above equations can be expressed as:

$$U_1^r a_{i1} + U_2^r a_{i2} - h_i^r V^r - f_i^{r-1} V^{r-1} < V^{r-1}/2^{s+\log n/\varepsilon + r - 1}, \tag{33}$$

$$U_1^r a_{n+k-1,1} + U_2^r a_{n+k-1,2} - h_{n+r-1}^r V^r - V^{r-1} = 0, \tag{34}$$

with $k = r$, for $i = 1$ to $n + r - 2$. Now both residues of the knapsack weights: a_{i1} and a_{i2}, are employed but another unknown U_2^r is introduced. If the attacker sets $V^r = q_1 m_2$, where m_2 is arbitrarily chosen, then there is a negligible probability of returning a superincreasing series by Lemma 2. If the attacker sets $V^r = q_1$, then some h_i^r's will always exist to match the $\log q_2$ most significant bits of $f_i^{r-1} V^{r-1}$ and $(n + r - 1)\log q_2$ bits of information from the fractions f_i^k are effectively lost. Then the security measure of (26) is again sufficient.

Acknowledgement

The research reported above was supported in part by an Operating Grant from the Natural Sciences and Engineering Research Council of Canada. We are grateful for helpful comments from thesis supervisors Lloyd Peppard and Stafford Tavares and also Carlyle Adams, Selim Akl, Ed Dawson, Hank Meijer, Brian O'Higgins, Paul Van Oorschot, Michael Wiener, and reviewers.

References

[1] W. Diffie and M. E. Hellman, "New directions in cryptography", *IEEE Trans. Inf. Theory*, vol. IT-22, pp. 644–654, Nov. 1976.

[2] E. F. Brickell and A. M. Odlyzko, "Cryptanalysis: A survey of recent results", *Proc. IEEE*, vol. 76, pp. 578–592, May 1988.

[3] E. F. Brickell, "The cryptanalysis of knapsack cryptosystems", in *Proc. of the 3rd SIAM Discrete Mathematics Conference*, Philadelphia, PA: SIAM, 1988, pp. 3 –23.

[4] Y. G. Desmedt, "What happened with knapsack cryptographic schemes?", in *Performance Limits in Communication Theory and Practice*, Kluwer Academic Publishers, 1988, pp. 113 –134.

[5] A. M. Odlyzko, "The rise and fall of knapsack cryptosystems", *Cryptology and Computational Number Theory*, in *Proc. Symp. Appl. Math.*, Am. Math. Soc., vol. 42, 1990, pp. 75 –80.

[6] R. C. Merkle and M. E. Hellman, "Hiding information and signatures in trapdoor knapsacks", *IEEE Trans. Inform. Theory*, vol. IT-24, no. 5, pp. 525 –530, Sept. 1978.

[7] R. M. Goodman and A. J. McAuley, "New trapdoor knapsack public key cryptosystem", *IEE Proceedings*, vol. 132, part E, no. 6, pp. 289 –292, Nov. 1985.

[8] D. E. Knuth, *The art of computer programming — volume 2 / seminumerical algorithms*, 2nd ed. Reading, MA: Addison-Wesley, 1981.

[9] C. Asmuth and J. Bloom, "A modular approach to key safeguarding", *IEEE Trans. Inform. Theory*, vol. IT-29, no. 2, pp. 208 –210, March 1983.

[10] M. R. Garey and D. S. Johnson, *Computers and intractability: A guide to the theory of NP - completeness*. San Francisco: W. H. Freeman and Co., 1979.

[11] B. Chor and R. L. Rivest, "A knapsack type public-key cryptosystem based on arithmetic in finite fields", *IEEE Trans. Inform. Theory*, vol. IT-34, no. 5, pp. 901 –909, 1988.

[12] W. A. Webb, "A public key cryptosystem based on complementing sets", *Cryptologia*, vol. XVI, no. 2, pp. 177–181, April 1992.

[13] Y. Desmedt, J. Vandewalle, and R. Govaerts, "The most general cryptographic knapsack scheme", *Proc. 1984 Carnahan Conf. on Security Technology*, New York: IEEE, 16 –18 May 1984, pp. 115 –120.

[14] J. Vyskoc, "Knapsack in cryptography", *Comput. Artif. Intell.*, vol. 6. no. 6., pp. 535 –40, 1987.

[15] C.–S. Laih, J. –Y. Lee, L. Harn, and Y. –K. Su., "Linearly shift knapsack public-key cryptosystem", *IEEE J. Sel. Areas Commun.*, vol. 7, no. 4, pp. 534 –539, May 1989.

[16] A. Shamir, "On the cryptocomplexity of knapsack systems", in *Proc. 11th ACM Symp. Theory Comput.*, 1979, pp. 118 –129.

[17] A. Shamir and R. E. Zippel, "On the security of the Merkle –Hellman cryptographic scheme", *IEEE Trans. Informat. Theory*, vol. IT-26, no. 3, pp. 339 –340, May 1980.

[18] R. Eier and H. Lagger, "Trapdoors in knapsack cryptosystems", *Cryptography*, Burg Feuerstein, Germany, March 29, 1982, Lecture Notes in Computer Science, vol. 149, Springer –Verlag, 1983, pp. 316 –322.

[19] E. F. Brickell and G. J. Simmons, "A status report on knapsack based public-key cryptosystems", Sandia Nat. Lab. Rep., 1983.

[20] Y. G. Desmedt, J. P. Vanderwalle and R. J. M. Govaerts, "A critical analysis of the security of knapsack public key algorithms", *IEEE Trans. Inform. Theory*, IT-30, no. 4, pp. 601 –11, July 1984.

[21] A. Shamir, "A polynomial time algorithm for breaking the basic Merkle –Hellman cryptosystem", *IEEE Trans. Inform. Theory*, vol. IT-30, no. 5, pp. 699 –704, Sept. 1984.

[22] J. C. Lagarias, "Knapsack public key cryptosystems and diophantine approximation", in *Advances in Cryptology CRYPTO '83*, New York: Plenum Press, 1984, pp. 3 –23.

[23] L. M. Adleman, "On breaking generalized knapsack public key cryptosystems", in *Proc. of the Fifteenth ACM Symp. Theory Comput.*, 1983, pp. 402 –412.

[24] E. F. Brickell, " Breaking iterated knapsacks", *CRYPTO '84*, Springer–Verlag, pp. 342 –358.

[25] J. Stern and P. Toffin, "Cryptanalysis of a public-key cryptosystem based on approximations by rational numbers", in *Advances in Cryptology EUROCRYPT '90*, Springer–Verlag, 1991, pp. 47 –55.

[26] P. van Emde Boas, "Another NP-complete partition problem and the complexity of computing short vectors in a lattice", Rept. 81 -04, Dept. of Mathematics, Univ. of Amsterdam, 1981.

[27] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovasz, "Factoring polynomials with rational coefficients", *Mathematische Annalen 261*, pp. 515 –534, 1982.

[28] C. P. Schnorr, "A hierarchy of polynomial time lattice basis reduction algorithms", *Theoretical Computer Science.*, vol. 53, pp. 201 –224, 1987.

[29] C. P. Schnorr, "An efficient algorithm for lattice basis reduction ", *J. Algorithms*, vol. 9, pp. 47 –62, 1988.

[30] E. F. Brickell, "Solving low density knapsacks", in *Advances in Cryptology CRYPTO '83*, New York: Plenum Press, 1984, pp. 25 –37.

[31] F. Jorissen, J. Vandewalle, and R. Govaerts, "Extension of Brickell's algorithm for breaking high density knapsacks", in *Advances in Cryptology EUROCRYPT '87*, 1988, pp. 109 –115.

[32] J. C. Lagarias and A. M. Odlyzko, "Solving low-density subset sum problems", *JACM*, vol. 32, no. 1, pp. 229–246, Jan. 1985.

[33] A. M. Odlyzko, "Cryptanalytic attacks on the multiplicative knapsack cryptosystem and on Shamir's fast signature scheme", *IEEE Trans. Inform. Theory*, vol. IT-30, no. 4, pp. 594 –601, 1984.

[34] A. M. Frieze, J. Hastad, R. Kannan, J. C. Lagarias, and A. Shamir, "Reconstructing truncated integer variables satisfying linear congruences", *SIAM J. Comput.*, vol. 17, pp. 262 –80, 1988.

[35] S. P. Radziszowski and D. L. Kreher, "Solving subset sum problems with the L^3 algorithm", *J. Combin. Math. Combin. Comput.*, vol. 3, pp. 49 –63, 1988.

[36] M. J. Coster, B. A. LaMacchia, A. M. Odlyzko, and C. P. Schnorr, "An improved low-density subset sum algorithm", in *Advances in Cryptology EUROCRYPT '91*, 1991, pp. 54 –67.

[37] A. Joux, and J. Stern, "Improving the critical density of the Lagarias –Odlyzko attack against knapsacks", in *Found. Comput. Theory, FCT 91, Lecture Notes in Comp. Sci.*, vol. 529, Springer –Verlag, pp. 258 –264.

[38] C. P. Schnorr, and M. Euchner, "Lattice basis reduction: Improved practical algorithms and solving subset sum problems", in *Foundations of Computation Theory, FCT 91, Lecture Notes in Computer Science*, vol. 529, New York: Springer –Verlag, pp. 68 –95.

[39] R. Schroeppel and A. Shamir, "A $TS^2 = O(2^n)$ time/space tradeoff for certain NP-complete problems", in *Proc. IEEE 20th Annual Symp. Found. Comp. Sci.*, Oct. 1979.

[40] A. G. Ferreira, "A parallel time/hardware tradeoff $T \cdot H = O(2^{n/2})$ for the knapsack problem", *IEEE Trans. Comput.*, vol. TC-40, no. 2, pp. 221 –225, Feb. 1991.

[41] H. W. Lenstra, Jr., "Integer programming with a fixed number of variables", *Math. Operations Research*, vol. 8, no. 4, pp. 538 –548, Nov. 1983.

[42] R. Kannan, "Improved algorithms for integer programming and related lattice problems", in *Proc. 15th Annual ACM Symp. Theory Comput.*, 1983, pp. 193 –206.

[43] H. R. Amirazizi, E. D. Karnin, and J. M. Reyneri, "Compact knapsacks are polynomially solvable", *ACM SIGACT NEWS*, vol. 15, pp. 20 –22, 1983.

[44] A. Shamir, "The cryptographic security of compact knapsacks", in *Proc. 1980 Symp. on Security and Privacy*, IEEE Computer Society, pp. 94 –99, April 1980.

[45] P. S. Henry, "Fast implementation of knapsack cipher", *Bell System Tech. J.*, vol. 60, pp. 767 –773, 1981.

[46] R. R. Jueneman, "Electronic document authentication", *IEEE Network*, vol. 1, no. 2, pp. 17 –23, 1987.

[47] G. A. Orton, "Very large scale arithmetic with applications to cryptography", Ph.D. thesis, Electrical Engineering, Queen's University, Kingston, Ontario, Canada, 1992.

[48] G. A. Orton, "A fast decoder for multiple-iterated knapsacks" in *17th Biennial Symposium on Communications*, Kingston, Ontario, Canada, May 1994, pp. 411 –416.