

Security in the presence of **decryption failures**

John Proos

University of Waterloo

August 13th 2003

Introduction

There exist public-key encryption schemes such as AD, GGH and NTRU for which valid ciphertexts do not always decrypt correctly.

Such schemes do not meet the definition of a public-key encryption scheme.

Suppose the decipherability of a ciphertext is dependent on the secret key. Then knowledge of correct or incorrect deciphering can leak information about the secret key.

Introduction con't

The NTRU cryptosystem is based on polynomial algebra modulo two distinct moduli.

At Crypto 2000 Jaulmes and Joux presented a chosen-ciphertext attack on NTRU using invalid ciphertexts.

To thwart such invalid ciphertext attacks various new padding schemes have been suggested for the NTRU cryptosystem.

Recently there have been attacks proposed against NTRU which exploit its decryption failures to recover the user's secret keys.

Outline

1. Introduction
2. Encryption schemes and security
3. The NTRU primitive
4. Attacks on the NTRU primitive
5. NTRU padding schemes and standards

Public-Key Encryption Schemes

A **public-key encryption scheme** (PKE) is a triple of algorithms, $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$, where

- \mathcal{K} , the **key generation algorithm**, returns pairs (pk, sk) of matching public and secret keys.
- \mathcal{E} , the **encryption algorithm**, takes pk and a message $x \in \mathcal{M}$ and returns a ciphertext $y \in \mathcal{C}$. Denoted $\mathcal{E}_{pk}(x) = y$ or $\mathcal{E}_{pk}(x; r) = y$.
- \mathcal{D} , the **decryption algorithm**, takes sk and a ciphertext y and returns either an $x \in \mathcal{M}$ or \perp to indicate that the ciphertext was invalid. Denoted $\mathcal{D}_{sk}(y) = x$ or \perp .

Imperfect PKE's

An **imperfect public-key encryption scheme (IPKE)** is a PKE without the requirement of perfect decryption.

That is for an IPKE there can exist key pairs (pk, sk) , messages $x \in \mathcal{M}$ and nonces r such that $\mathcal{E}_{pk}(x; r) = y$ but $\mathcal{D}_{sk}(y) \neq x$.

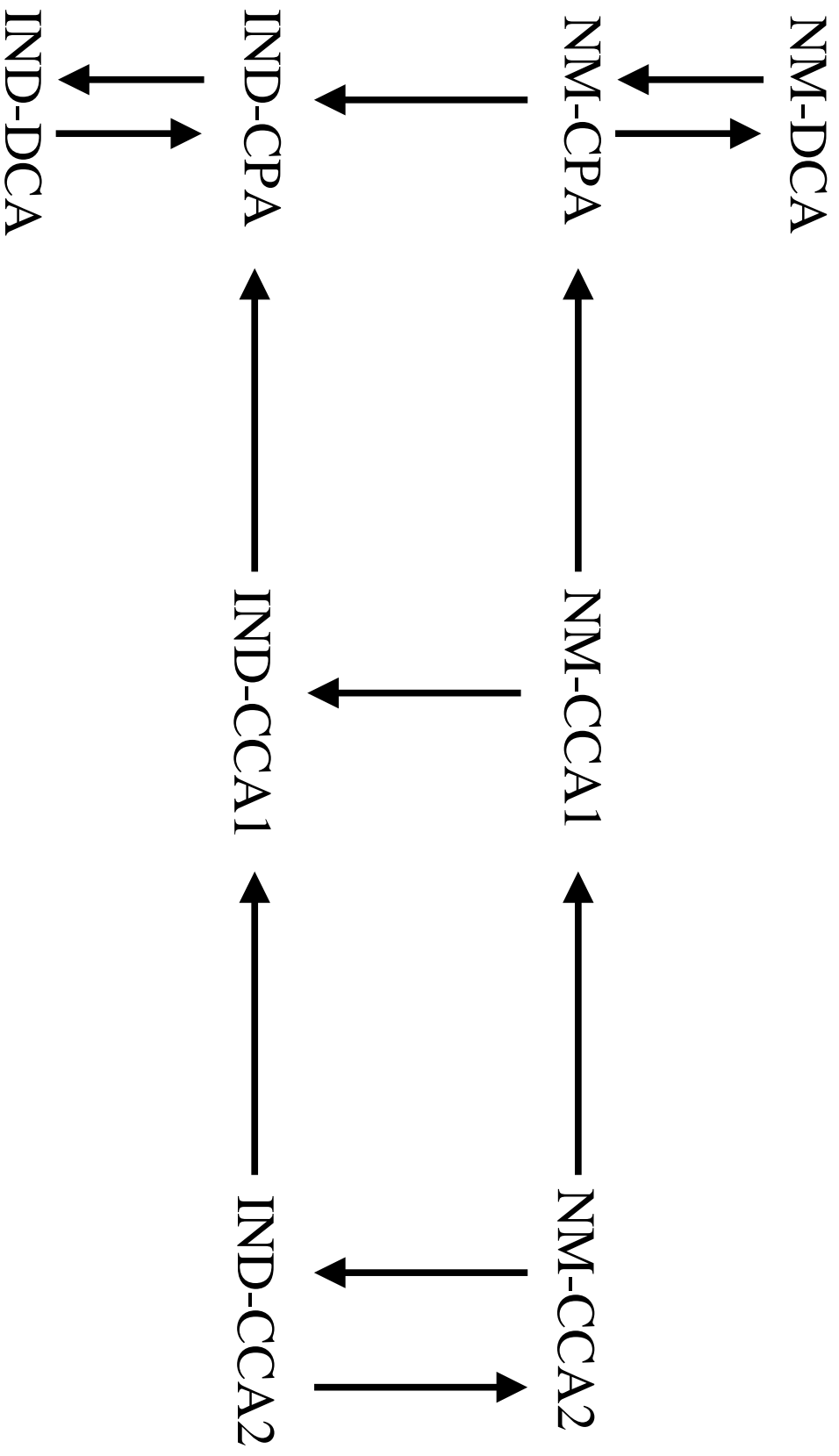
A valid ciphertext y for $x \in \mathcal{M}$ which does not decrypt to x will be called **indecipherable** with respect to x .

Security Notions

- **One-way (OW):** Given $y = \mathcal{E}_{pk}(x; r)$ an attacker can not recover the entire message x .
- **Indistinguishability (IND):** If an attacker selects $x_1, x_2 \in \mathcal{M}$ and someone selects $b \in \{1, 2\}$ and encrypts x_b to form y then the attacker can not guess the value of b with probability significantly better than one half.
- **Non-malleability (NM):** Given $y = \mathcal{E}_{pk}(x; r)$ an attacker can not form a new ciphertext y' whose message is related to x in any meaningful way.

Attacks on PKE's and IPKE's

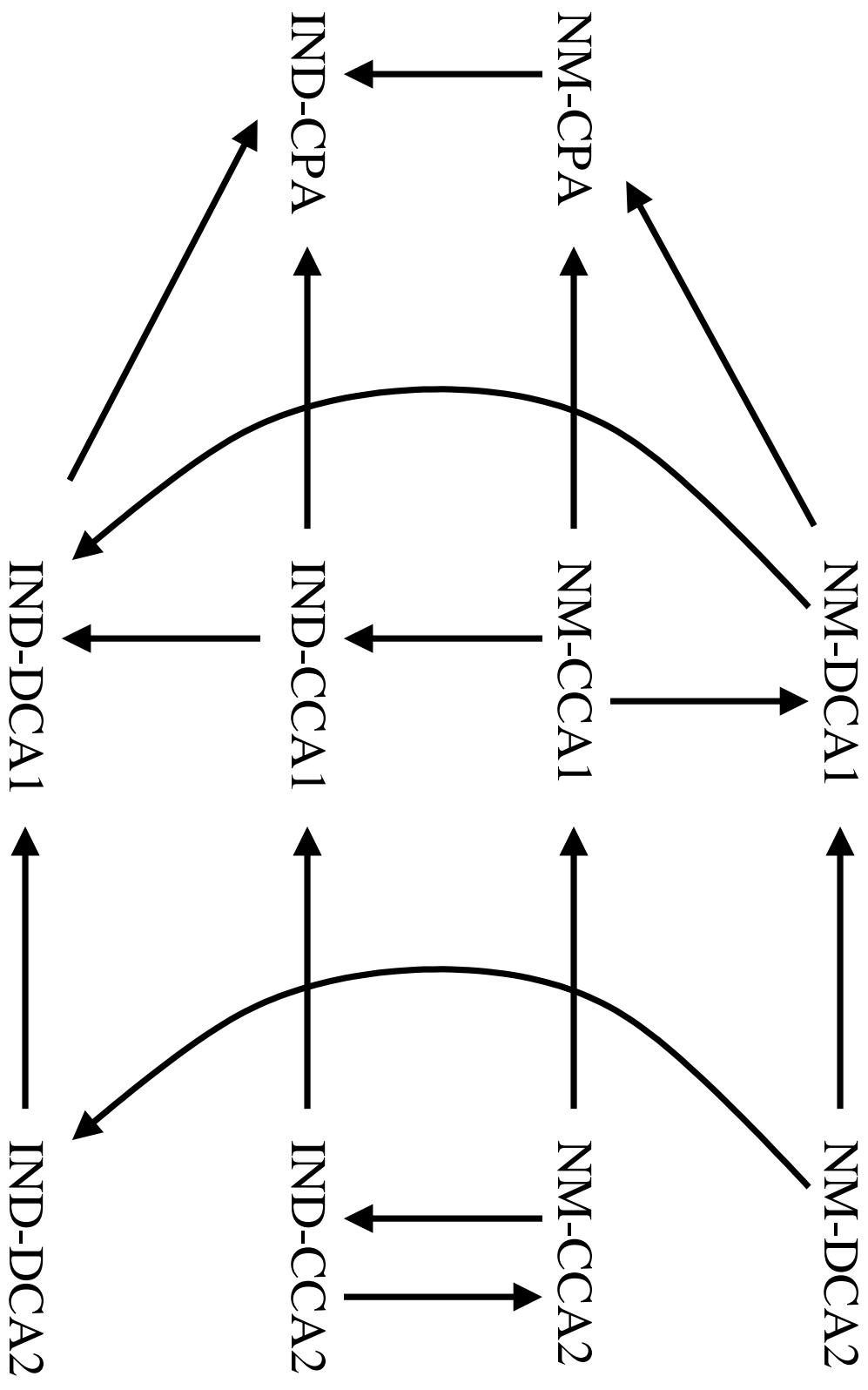
- **Chosen plaintext attacks (CPA):** The attacker has access to the public key and the encryption function.
- **Chosen-ciphertext attacks (CCA):**
The attacker is given an oracle that given y returns $D_{sk}(y)$.
- **Reaction attacks (RA):** The attacker is given an oracle that given y will return whether or not $\mathcal{D}_{sk}(y) = \perp$.
- **Decipherable ciphertext attacks (DCA):**
The attacker is given an oracle which given $x \in \mathcal{M}$, a nonce r and $y = \mathcal{E}_{pk}(x; r)$ returns whether or not $\mathcal{D}_{sk}(y) = x$.



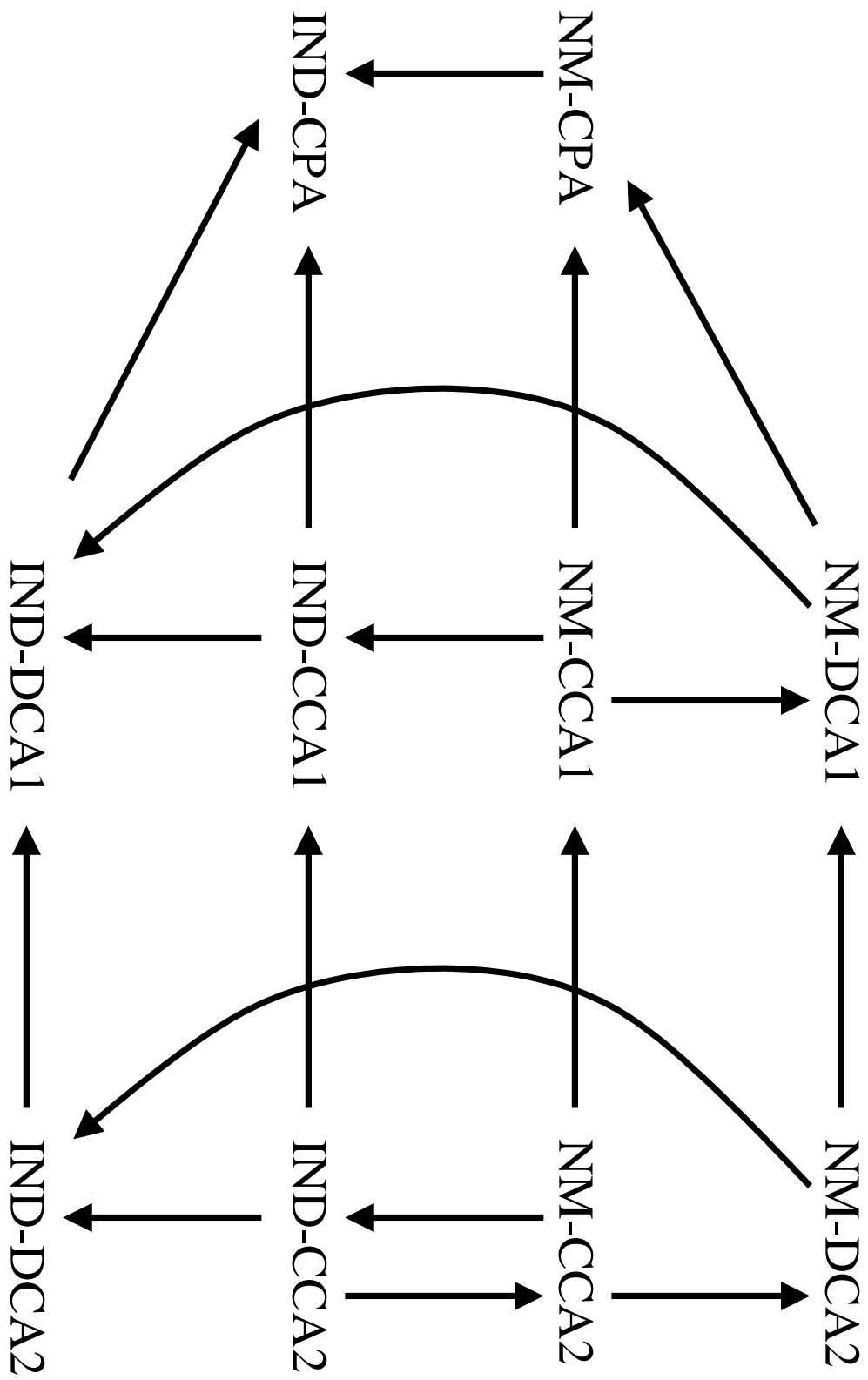
PKE Security Relations

Attacks on IPKE's vs PKE's

- **CPA:** No difference.
- **RA:** The attacker can determine if a valid ciphertext decrypts to a message or to \perp .
- **CCA:** The attacker can determine if a valid ciphertext is indecipherable and if so determine its decryption.
- **DCA:** Equivalent to a CPA attack against a PKE.



IPKE Relations 1



IPKE Relations 2

The NTRU Cryptosystem

Parameters:

- Integers N , q and $p = 3$
- Four sets of polynomials, \mathcal{L}_f , \mathcal{L}_g , \mathcal{L}_r and \mathcal{L}_m

$$\text{in } \mathbf{R} = \mathbb{Z}[\mathbf{X}]/(\mathbf{X}^N - 1)$$

$$\mathcal{L}_m = \left\{ m \in \mathbf{R} : m \text{ has coefficients in } \{-1, 0, 1\} \right\}$$

\mathcal{L}_f , \mathcal{L}_g and \mathcal{L}_r are also $\{-1, 0, 1\}$ polynomials, but with a specified number of 1's and -1 's

NTRU Key Generation

Select $f \in \mathcal{L}_f$ and $g \in \mathcal{L}_g$ such that there exists f_p^{-1} and f_q^{-1} for which

$$f \cdot f_p^{-1} \equiv 1 \pmod{p}$$

$$f \cdot f_q^{-1} \equiv 1 \pmod{q}$$

The **public key** is $h \equiv f_q^{-1} \cdot g \pmod{q}$.

The **secret key** is (f, f_p^{-1}) .

NTRU Encryption

Given a public key h and message $m \in \mathcal{L}_m$.
Select a random element $r \in \mathcal{L}_r$ and calculate

$$e = m + prh \pmod{q}$$

NTRU Decryption

Given $e = m + prh \pmod{q}$.

Calculate

$$\begin{aligned} a &\equiv ef \pmod{q} \\ &\equiv mf + prhf \pmod{q} \\ &\equiv mf + prg \pmod{q} \end{aligned}$$

Reducing a into the range $(-q/2, q/2]$.

Assume that $a = mf + prg$ and calculate

$$\begin{aligned} af_p^{-1} &\equiv (mf + prg)f_p^{-1} \pmod{p} \\ &\equiv m \pmod{p} \end{aligned}$$

Imperfect Decryption

If $mf + prg \pmod{q} \neq mf + prg$ then the decryption algorithm will fail to produce m .

Thus the NTRU cryptosystem has imperfect decryption.

Fact: For the suggested NTRU parameters the indecipherable valid ciphertexts uniquely determine the secret key.

Suggested Parameters with $p = 3$

The following parameters were suggested in NTRU technical report #12.

N	q	\mathcal{L}_f	\mathcal{L}_g	\mathcal{L}_r
107	64	$\mathcal{L}(15, 14)$	$\mathcal{L}(12, 12)$	$\mathcal{L}(5, 5)$
167	128	$\mathcal{L}(61, 60)$	$\mathcal{L}(20, 20)$	$\mathcal{L}(18, 18)$
263	128	$\mathcal{L}(50, 49)$	$\mathcal{L}(24, 24)$	$\mathcal{L}(16, 16)$
503	256	$\mathcal{L}(216, 215)$	$\mathcal{L}(72, 72)$	$\mathcal{L}(55, 55)$

where

$$\mathcal{L}(d_1, d_{-1}) = \left\{ z \in R : \begin{array}{l} z \text{ has } d_1 \text{ coefficients} \\ \text{equal to } 1, d_{-1} \text{ equal} \\ \text{to } -1 \text{ and the rest } 0 \end{array} \right\}$$

Outline of Attack 1 on NTRU

This attack assumes the attacker has access to an oracle which given (m, r) and $e = m + prh \pmod{q}$ determines if the decryption algorithm will decrypt e to m .

Phase 1: Find a pair $(m, r) \in \mathcal{L}_m \times \mathcal{L}_r$ which generate an indecipherable valid ciphertext.

Phase 2: Use the pair (m, r) to find a pair (m', r') which generates a ciphertext which is 'just' indecipherable.

Phase 3: Use (m', r') to determine f .

Running Times of Attack 1

The attack was run against 100 instances of NTRU for each of the four sets of suggested parameters with the following results.

N	Oracle Calls		Avg Indec Calls
	Average	Phase 1 Range	
107	26241	106 – 163844	222.35
167	20105	62 – 116576	218.5
263	822979	10632 – 4357309	505.6
503	27259	173 – 165589	602.01

Outline of Attack 2 on NTRU

The attacker is given access to a DC oracle.

It is also assumed that each time a message is encrypted the r must be selected at random.

Phase 1: Find a pair $(m, r) \in \mathcal{L}_m \times \mathcal{L}_r$ which generate an indecipherable valid ciphertext.

Phase 2: Keep m fixed and use the oracle to determine a list of r values for which $\mathcal{E}(m; r)$ does not decrypt to m .

Phase 3: Use the distributions of 1's and -1 's in the r 's found in phase 2 to determine g .

Basis of Attack 2 on NTRU

If $mf + prg \pmod{q} \neq mf + prg$ it is because it has a coefficient outside $(-q/2, q/2]$.

Goal of phase 1: Determine a message m for which mf has one coefficient which is larger in absolute value than all the others.

There will then be a correlation between the r 's found in phase 2 and g from which g can be determined.

Running Times of Attack 2

Results when the attack was run against 100 instances of NTRU for each parameter set.

N	Iterations for Success							Iteration
	1	2	3	4	5	6	7	Suc Rate
107	65	26	7	2	0	0	0	68.5%
167	76	15	6	2	1	0	0	73.0%
263	66	18	12	2	1	1	0	63.7%
503	47	24	18	7	2	1	1	50.0%

The maximum number of phase 2 oracle calls were 500,000 for $N = 107, 167$ and 3,000,000 for $N = 263, 503$.

Phase 2 results over successful iterations.

N	Oracle Calls	Avg Indec Calls		Avg
	Average	Num	Rate	Gap
107	134809	34.60	3896	4.60
167	110519	62.45	1770	11.02
263	575804	69.75	8255	13.18
503	565486	323.00	1751	20.92

Padding Schemes

Prior to DCA Attacks

NTRU Cryptosystems Inc. had suggested three padding schemes for NTRU.
(NTRU-I, NTRU-II and NTRU-III).

Nguyen and Pointcheval suggested three new padding schemes - two based on REACT and the other on OAEP - and analyzed the security of all six padding schemes.

Post DCA Attacks

NTRU Cryptosystems Inc. have proposed a new padding scheme called NAEP.

Attacks on the Padding Schemes

The six pre-DCA attack padding schemes are all not OW-DCA secure.

The first REACT based scheme can be broken using the first DCA attack.

The second REACT based scheme can be broken using the second DCA attack.

The other pre-DCA attack padding schemes can be broken using variations of the second DCA attack.

The NAEP Padding Scheme

All existing DCA attacks on NTRU require the attacker to have control over $m \in \mathcal{L}_m$ and/or $r \in \mathcal{L}_r$.

The NAEP is designed to prevent an attacker from controlling the $m \in \mathcal{L}_m$ and the $r \in \mathcal{L}_r$ used during encryption.

Note that NAEP does not prevent DCA attacks, it simply makes them more difficult.

The NAEP Padding Scheme Con't

NAEP uses two hash functions, denoted H and G and encrypts a plaintext S as follows:

Select a random string T and derive $r \in \mathcal{L}_r$ from $H(S||T)$.

Calculate $z = prh \pmod{q}$ and derive $m \in \mathcal{L}_m$ from $S||T \oplus G(z)$.

Set the ciphertext to be $e = m + prh \pmod{q}$.

EESS#1 Version 1.0

The “Consortium for Efficient Embedded Security” proposed a standard, **EESS#1**, for NTRU.

Version 1.0 of the standard used a version of NTRU with $p = X + 2$ and the NTRU-III padding scheme.

Version 1.0 contained parameter sets for four security levels. However, the secret keys could be recovered using DCA attacks.

EESS#1 Version 2.0

Version 2.0 appeared in June and uses the new NAEP padding scheme.

Version 2.0 has a single parameter set that has $p = 2$.

The security of the version 2.0 parameter set is based on the fact that to an attacker the scheme is indistinguishable from a PKE.

There are no known DCA attacks which can recover secret keys for the EESS#1 version 2.0 parameter set.

Conclusions

- Security against DCA attacks must be considered for any encryption scheme without perfect decryption.
- It is very dangerous to attempt to apply security results for PKE's to encryption schemes without perfect decryption.
- The parameters suggested for NTRU, with $p = 3$ and $p = X + 2$ are not sufficient to protect the scheme from DCA attacks.
- Parameters for NTRU should be selected in such a way that finding an indecipherable valid ciphertext is intractable.