RabinUnique/Another attack on RSA/LWE-KE/DH revisited

Another way to make Rabin Unique

Recall Rabin's Encryption Scheme

n is a security parameter

1. Alice gen p, q primes of length n. Let $N = pq$. Send N.
2. Encode: To send m, Bob sends $c = m^2 \pmod{N}$.
3. Decode: Alice can find m such that $m^2 \equiv c \pmod{N}$.

Recall Rabin's Encryption Scheme

n is a security parameter

1. Alice gen p, q primes of length n. Let $N = pq$. Send N.
2. Encode: To send m, Bob sends $c = m^2 \pmod{N}$.
3. Decode: Alice can find m such that $m^2 \equiv c \pmod{N}$. OH! There will be two or four of them! What to do?

Making Rabin Unique. We call it RabinU

n is a security parameter

1. Alice gen p, q primes of length n. Let $N = pq$. NEW: Let x be a rand element of NSQ_N. Send (N, x).
2. Encode: To send m, Bob sends
   2.1 $c = m + xm^{-1} \pmod N$,
   2.2 0 if $m \in SQ_N$, 1 if $m \in NSQ_N$, and
   2.3 0 if $(cm^{-1} \mod N > m)$, 1 if $(cm^{-1} \mod N < m)$.
3. Decode: Alice needs m st $c = m + xm^{-1}$, so solve $m^2 - cm + c = 0$. This gives 2 or 4 roots. The info about $m \in SQ_N$ and $cm^{-1} \mod N < m$. uniquely determines which root. (We skip details)

Note: RabinU can be cracked iff Factoring is easy.

Yet Another RSA attack

Review of RSA Attacks

1. If N is small, Eve Factors. Response: Use p, q large.
2. If same e, $e \leq L$. Low-e attack. Response: Large e.
3. If same e, $m^e < N_1 \cdots N_L$. Low-e attack. Response: Pad m.
4. NY,NY problem. Leaks info. Response: Rand Pad m
5. Timing Attacks: Response: Rand Pad time.

Note items 2 and 3:

e same but N's Different

How about

N same but e's Different

Surely that can't be a problem!

Review of RSA Attacks

1. If N is small, Eve Factors. Response: Use p, q large.
2. If same e, $e \leq L$. Low-e attack. Response: Large e.
3. If same e, $m^e < N_1 \cdots N_L$. Low-e attack. Response: Pad m.
4. NY,NY problem. Leaks info. Response: Rand Pad m
5. Timing Attacks: Response: Rand Pad time.

Note items 2 and 3:

e same but N's Different

How about

N same but e's Different

Surely that can't be a problem!

Or can it!

Review of RSA Attacks

1. If N is small, Eve Factors. Response: Use p, q large.
2. If same e, $e \leq L$. Low-e attack. Response: Large e.
3. If same e, $m^e < N_1 \cdots N_L$. Low-e attack. Response: Pad m.
4. NY,NY problem. Leaks info. Response: Rand Pad m
5. Timing Attacks: Response: Rand Pad time.

Note items 2 and 3:

e same but N's Different

How about

N same but e's Different

Surely that can't be a problem!

Or can it!

Won't bother with a vote, onto the next slide.

Same N, Different e, Eve Cracks RSA

1. Alice gives B_1 (N, e_1)
2. Alice gives B_2 (N, e_2)
3. e_1, e_2 are rel prime (Bad idea?).

Alice sends m to both B_1 and B_2. Eve sees

1. $m^{e_1} \pmod{N}$
2. $m^{e_2} \pmod{N}$

Same N, Different e, Eve Cracks RSA

1. Alice gives B_1 (N, e_1)
2. Alice gives B_2 (N, e_2)
3. e_1, e_2 are rel prime (Bad idea?).

Alice sends m to both B_1 and B_2. Eve sees

1. $m^{e_1} \pmod{N}$
2. $m^{e_2} \pmod{N}$

e_1, e_2 rel prime, so $\exists\, x, y \in \mathbb{Z}$ $e_1 x + e_2 y = 1$.

Same N, Different e, Eve Cracks RSA

1. Alice gives B_1 (N, e_1)
2. Alice gives B_2 (N, e_2)
3. e_1, e_2 are rel prime (Bad idea?).

Alice sends m to both B_1 and B_2. Eve sees

1. $m^{e_1} \pmod N$
2. $m^{e_2} \pmod N$

e_1, e_2 rel prime, so $\exists\, x, y \in \mathbb{Z}$ $e_1 x + e_2 y = 1$. Eve finds x, y with Euclidean Algorithm and then:

$$(m^{e_1})^x \times (m^{e_2})^y \pmod N = m^{e_1 x + e_2 y} \pmod N = m \pmod N$$

Caveat: if (say) $x < 0$ need m^{e_1} to have inverse mod N.
Note: Eve found m without factoring N.
Response: Use Different N.

Advice for Alice When she uses RSA

Alice will use RSA with people $A_1, \ldots, A_L$. Will use $(N_i = p_i q_i, e_i)$ for A_i.

1. Pick p_i, q_i large and different.
2. Can have all e_i's the same e but should be large.
3. Randomly Pad m
4. Randomly pad time

Advice for Alice When she uses RSA

Alice will use RSA with people $A_1, \ldots, A_L$. Will use $(N_i = p_i q_i, e_i)$ for A_i.

1. Pick p_i, q_i large and different.
2. Can have all e_i's the same e but should be large.
3. Randomly Pad m
4. Randomly pad time

Same e?: Good idea or bad idea? Will consider on Wedensday.

Key Exchange With Matrices and Lattices

DH and RSA Rely on Number Theory

(We are revisiting the guest lecture on this topic.)

1. DH and RSA rely on problems in Number Theory being hard.
2. If DL is easy then DH is cracked (not conversely).
3. If Factoring is easy then RSA is cracked (not conversely).
4. DL and Factoring are in Quantum-P (QP).
5. If Quantum Computers (QC) ever become a reality than DH and RSA are cracked!

How worried should we be? Discuss

Is QC Really a Threat?

My opinion

1. QCs seem hard to build.
2. I do not work in either QC; I have no special insights.
3. QC is worth studying for the insight it gives into both quantum and computing.
4. There are classical algorithms for DL and factoring that are forcing crypto people to up their game.

Is QC Really a Threat?

My opinion

1. QCs seem hard to build.
2. I do not work in either QC; I have no special insights.
3. QC is worth studying for the insight it gives into both quantum and computing.
4. There are classical algorithms for DL and factoring that are forcing crypto people to up their game.

Final Opinion: Studying public-key crypto that does not depend on number theory assumptions is intellectually awesome. Might not be needed for QC, but perhaps for other scenarios.

Post-Quantum Cryptography

This is a great title since

1. It has nothing to do with Quantum, so its not that hard.
2. It sounds cool and can attract funding.

It just means that we are not using number-theory assumptions.

Small Vectors

Definition

Assume $n \in \mathbb{N}$ and p is a prime. Pick a random small $\vec{e} \in \mathbb{Z}_p^n$ means pick each component as a discrete Gaussian with mean 0 and variance to be specified.

LWE Key Exchange (Due to Regev)

LWE Key Exchange: From now on LWE-KE
LWE means Learning With Errors. We will not need this.

1. We will discuss the protocol and how it works.
2. We will discuss hardness assumptions later.

LWE-KE. Two Security Parameters n, n'

1. Alice: rand prime p of length n', rand $n \times n$ matrix A over $\mathbb{Z}_p$.
2. Alice: rand $\vec{y} \in \mathbb{Z}_p^n$, small $\vec{e}_y \in \mathbb{Z}_p^n$. Sends $\vec{y}A + \vec{e}_y$.
3. Bob: rand $\vec{x} \in \mathbb{Z}_p^n$, small $\vec{e}_x \in \mathbb{Z}_p^n$. Sends $A\vec{x} + \vec{e}_x$.
4. Alice computes $a = \vec{y}(A\vec{x} + \vec{e}_x) = \vec{y}A\vec{x} + \vec{y} \cdot \vec{e}_x$.
5. Bob computes $b = (\vec{y}A + \vec{e}_y)\vec{x} = \vec{y}A\vec{x} + \vec{x} \cdot \vec{e}_y$.
6. They share $\vec{y}A\vec{x}$

LWE-KE. Two Security Parameters n, n'

1. Alice: rand prime p of length n', rand $n \times n$ matrix A over $\mathbb{Z}_p$.
2. Alice: rand $\vec{y} \in \mathbb{Z}_p^n$, small $\vec{e}_y \in \mathbb{Z}_p^n$. Sends $\vec{y}A + \vec{e}_y$.
3. Bob: rand $\vec{x} \in \mathbb{Z}_p^n$, small $\vec{e}_x \in \mathbb{Z}_p^n$. Sends $A\vec{x} + \vec{e}_x$.
4. Alice computes $a = \vec{y}(A\vec{x} + \vec{e}_x) = \vec{y}A\vec{x} + \vec{y} \cdot \vec{e}_x$.
5. Bob computes $b = (\vec{y}A + \vec{e}_y)\vec{x} = \vec{y}A\vec{x} + \vec{x} \cdot \vec{e}_y$.
6. They share $\vec{y}A\vec{x}$

Hey! That does not make sense! Neither one has $\vec{y}A\vec{x}$!

LWE-KE

Alice has $a = \vec{y}(A\vec{x} + \vec{e}_x) = \vec{y}A\vec{x} + (\vec{y} \cdot \vec{e}_x)$.

Bob has $b = (\vec{y}A + \vec{e}_y)\vec{x} = yA\vec{x} + (\vec{x} \cdot e_y)$.

Since $\vec{e}_x, \vec{e}_y$ small, $a \sim b$.

LWE-KE

Alice has $a = \vec{y}(A\vec{x} + \vec{e}_x) = \vec{y}A\vec{x} + (\vec{y} \cdot \vec{e}_x)$.

Bob has $b = (\vec{y}A + \vec{e}_y)\vec{x} = yA\vec{x} + (\vec{x} \cdot e_y)$.

Since $\vec{e}_x, \vec{e}_y$ small, $a \sim b$.

SO WHAT! $a \sim b$??? What does $\sim$ even mean over $\mathbb{Z}_p$? What kind of DELETED – WE ARE BEING TAPED is this? Discuss

LWE-KE

Alice has $a = \vec{y}(A\vec{x} + \vec{e}_x) = \vec{y}A\vec{x} + (\vec{y} \cdot \vec{e}_x)$.

Bob has $b = (\vec{y}A + \vec{e}_y)\vec{x} = yA\vec{x} + (\vec{x} \cdot e_y)$.

Since $\vec{e}_x, \vec{e}_y$ small, $a \sim b$.

SO WHAT! $a \sim b$??? What does $\sim$ even mean over $\mathbb{Z}_p$? What kind of DELETED – WE ARE BEING TAPED is this? Discuss

CALM DOWN! If pick variance cleverly then with high prob either

$a, b \in \{0, 1, 2, \ldots, p/4\} \cup \{3p/4, \ldots, p-1\}$ ("close to 0"), OR

$a, b \in \{p/4 + 1, \ldots, 3p/4 - 1\}$ ("close to $p/2$")

(Paper with this on course website under notes.)

LWE-KE. Two Security Parameters n, n'

1. Alice: rand prime p of length n', rand $n \times n$ matrix A over $\mathbb{Z}_p$.
2. Alice: rand $\vec{y} \in \mathbb{Z}_p^n$, small $\vec{e}_y \in \mathbb{Z}_p^n$. Sends $\vec{y}A + \vec{e}_y$.
3. Bob: rand $\vec{x} \in \mathbb{Z}_p^n$, small $\vec{e}_x \in \mathbb{Z}_p^n$. Sends $A\vec{x} + \vec{e}_x$.
4. Alice computes $a = \vec{y}(A\vec{x} + \vec{e}_x) = \vec{y}A\vec{x} + \vec{y} \cdot \vec{e}_x$. If $a \in \{0, \ldots, p/4\} \cup \{3p/4, \ldots, p-1\}$, $s_A = 0$, else $s_A = 1$.
5. Bob computes $b = (\vec{y}A + \vec{e}_y)\vec{x} = \vec{y}A\vec{x} + \vec{x} \cdot \vec{e}_y$. If $b \in \{0, \ldots, p/4\} \cup \{3p/4, \ldots, p-1\}$, $s_B = 0$, else $s_B = 1$.
6. With high prob $s_A = s_B$. That is the bit they share.

PRO: Hardness Assumption NOT number-theoretic (next slide)
CON: Only 1 bit.

LWE-KE. HA

Definition

LWE (Learning with Errors) problem p a prime, $n \in \mathbb{N}$. $\vec{u} \in \mathbb{Z}_p^n$ is unknown. We want to learn $\vec{u}$. Our only operation is to

1. Pick a random $\vec{v} \in \mathbb{Z}_p^n$
2. Pick a random $e \in \mathbb{R}$ small
3. We get to ask for $(\vec{v}, \vec{v} \cdot \vec{u} + e)$

Solving LWE quickly means learning $\vec{u}$ with high prob after a poly (in n) number of operations.

Definition

GAP-SVP is a variant of Shortest Vector Problem.

Known: If can crack LWE-KE then can solve LWE.
Known: If can solve LWE then can crack GAP-SVP problem.
Upshot: If can crack LWE-KE then can solve GAP-SVP problem.
Caveat: The sense of can solve is odd- next slides.

LWE-KE. Hardness Assumption – A Caveat

We claimed:

$$\text{GAP-SVP} \leq \text{LWE} \leq \text{LWE-KE}$$

LWE-KE. Hardness Assumption – A Caveat

We claimed:

$$\text{GAP-SVP} \leq \text{LWE} \leq \text{LWE-KE}$$

This is true. Sort of.

LWE-KE. Hardness Assumption – A Caveat

We claimed:

$$\text{GAP-SVP} \leq \text{LWE} \leq \text{LWE-KE}$$

This is true. Sort of.
It uses Quantum Reductions.

LWE-KE. Hardness Assumption – A Caveat

We claimed:

$$\text{GAP-SVP} \leq \text{LWE} \leq \text{LWE-KE}$$

This is true. Sort of.
It uses Quantum Reductions.

1. QC: DH cracked, LWE-KE uncrackable if GAP-SVP hard.
2. $\neg QC$: DH looks save, LWE-KE crackability unknown.

LWE-KE. Hardness Assumption – A Caveat

We claimed:

$$\text{GAP-SVP} \leq \text{LWE} \leq \text{LWE-KE}$$

This is true. Sort of.
It uses Quantum Reductions.

1. QC: DH cracked, LWE-KE uncrackable if GAP-SVP hard.
2. $\neg QC$: DH looks save, LWE-KE crackability unknown.

My Opinion: LWE-KE looks uncrackable anyway. With enough fine tuning and improvements perhaps it could give RSA a run for its money! (And there is LOTS of money involved! Not quite related – check out the two music videos on the course website (1) Its all about the Benjamins, and (2) Its all about the Pentiums.)

LWE-KE. Practical Considerations

1. There is a version of LWE-KE where small means all components in

$$\{0, 1, -1\}$$

   where each picked with prob $1/3$. Note that -1 is $p - 1$.
2. There is a version of LWE-KE where many bits shared.
3. For both of the above version you gain efficiency but loose security guarantees.
4. Probably still secure.

Correction to Diffie-Helman

Recall the Diffie-Helman Key Exchange

1. Alice: rand (p, g), p of length n, g gen for $\mathbb{Z}_p$. Arith mod p.
2. Alice sends (p, g) to Bob in the clear (Eve can see it).
3. Alice: rand $a \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$, sends g^a.
4. Bob: rand $b \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$, sends g^b.
5. Alice:$(g^b)^a = g^{ab}$. Bob:$(g^a)^b = g^{ab}$. g^{ab} is shared secret.

Why does Alice: rand $a \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$.
Why not $a \in \{0, \ldots, p-1\}$? Discuss

Recall the Diffie-Helman Key Exchange

1. Alice: rand (p, g), p of length n, g gen for $\mathbb{Z}_p$. Arith mod p.
2. Alice sends (p, g) to Bob in the clear (Eve can see it).
3. Alice: rand $a \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$, sends g^a.
4. Bob: rand $b \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$, sends g^b.
5. Alice:$(g^b)^a = g^{ab}$. Bob:$(g^a)^b = g^{ab}$. g^{ab} is shared secret.

Why does Alice: rand $a \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$.
Why not $a \in \{0, \ldots, p-1\}$? Discuss
If g is small and a is small then Eve can determine a from g^a.

Recall the Diffie-Helman Key Exchange

1. Alice: rand (p, g), p of length n, g gen for $\mathbb{Z}_p$. Arith mod p.
2. Alice sends (p, g) to Bob in the clear (Eve can see it).
3. Alice: rand $a \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$, sends g^a.
4. Bob: rand $b \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$, sends g^b.
5. Alice:$(g^b)^a = g^{ab}$. Bob:$(g^a)^b = g^{ab}$. g^{ab} is shared secret.

Why does Alice: rand $a \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$.
Why not $a \in \{0, \ldots, p-1\}$? Discuss
If g is small and a is small then Eve can determine a from g^a.
But: Eve can compute $g^0, g^1, \ldots, g^L$ and if she sees any of those she knows.

Example

$p = 1013$
$g = 5$
$a = 6$
Eve computes ahead of time:
$5^0 = 1$
$5^1 = 5$
$5^2 = 25$
$5^3 = 125$
$5^4 = 625$
$5^5 = 86$
$5^6 = 430$
If Eve sees Alice 430 then she knows $a = 6$
Nothing special about a being small.

Example

$p = 1013$

$g = 40$

$a \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\} = \{337, \ldots, 674\}$

Note: We assume that Eve KNOWS these endpoints.

Eve computes

$40^{337} \equiv 919$

$40^{338} \equiv 292$

$40^{339} \equiv 537$

$40^{340} \equiv 207$

$40^{341} \equiv 176$

$40^{342} \equiv 962$

$40^{343} \equiv 999$

If Eve sees Alice send any of $919, 292, 537, 207, 176, 962, 999$ then she knows a

g was big, a was big. Didn't help!

Example

$p = 1013$

$g = 40$

$a \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\} = \{337, \ldots, 674\}$

Note: We assume that Eve KNOWS these endpoints.

Eve computes

$40^{337} \equiv 919$

$40^{338} \equiv 292$

$40^{339} \equiv 537$

$40^{340} \equiv 207$

$40^{341} \equiv 176$

$40^{342} \equiv 962$

$40^{343} \equiv 999$

If Eve sees Alice send any of $919, 292, 537, 207, 176, 962, 999$ then she knows a

g was big, a was big. Didn't help!

Of course, Eve has to get VERY LUCKY.

The Real Diffie-Helman

1. Alice finds a (p, g), p of length n, g gen for $\mathbb{Z}_p$. Arith mod p.
2. Alice sends (p, g) to Bob in the clear (Eve can see it).
3. Alice: rand $a \in \{0, \ldots, p-1\}$, sends g^a.
4. Bob: rand $b \in \{0, \ldots, p-1\}$, sends g^b.
5. Alice:$(g^b)^a = g^{ab}$. Bob:$(g^a)^b = g^{ab}$. g^{ab} is shared secret.

Eve comp $g^0, g^1, \ldots, g^L$. If $a \in \{0, \ldots, L\}$ Eve knows a.
Not really a problem:
Either

1. If L is small then Eve would have to get LUCKY to find a.
2. If L is large then Eve is doing LOTS OF computation.

Upshot: a, g small did not make attack much easier for Eve.

Is There Harm In Restricting a, b?

Have shown that requiring $a, b \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$ won't help.

Is There Harm In Restricting a, b?

Have shown that requiring $a, b \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$ won't help.

Will it hurt?
Vote: restricting a, b will

1. make DH less secure
2. not have any affect.

Is There Harm In Restricting a, b?

Have shown that requiring $a, b \in \{\frac{p}{3}, \ldots, \frac{2p}{3}\}$ won't help.

Will it hurt?
Vote: restricting a, b will

1. make DH less secure
2. not have any affect.

(1) Make DH less secure.
Key space is smaller, making it easier for Eve.

How Important Is Public Key?

Used Everywhere

Public key is mostly used for giving out keys to be used for classical systems.
This makes the following work:

Used Everywhere

Public key is mostly used for giving out keys to be used for classical systems.
This makes the following work:

1. Amazon – Credit Cards

Used Everywhere

Public key is mostly used for giving out keys to be used for classical systems.
This makes the following work:

1. Amazon – Credit Cards
2. Ebay – Paypal

Used Everywhere

Public key is mostly used for giving out keys to be used for classical systems.
This makes the following work:

1. Amazon – Credit Cards
2. Ebay – Paypal
3. Facebook privacy –

Used Everywhere

Public key is mostly used for giving out keys to be used for classical systems.
This makes the following work:

1. Amazon – Credit Cards
2. Ebay – Paypal
3. Facebook privacy – just kidding, Facebook has no privacy.

Used Everywhere

Public key is mostly used for giving out keys to be used for classical systems.
This makes the following work:

1. Amazon – Credit Cards
2. Ebay – Paypal
3. Facebook privacy – just kidding, Facebook has no privacy.
4. Every financial institution in the world.

Used Everywhere

Public key is mostly used for giving out keys to be used for classical systems.
This makes the following work:

1. Amazon – Credit Cards
2. Ebay – Paypal
3. Facebook privacy – just kidding, Facebook has no privacy.
4. Every financial institution in the world.
5. Military – though less is known about this.

Turing Awards

The Turing Award is The Nobel Prize of Computer Science.

Given out every year.

We note when someone mentioned in Public Key Crypto won.

1. 1976- Michael Rabin
2. 1995- Manuel Blum
3. 2002- Ron Rivest, Shamir, Len Adelman
4. 2012- Silvio Micali, Shaffi Goldwasser
5. 2015- Whitfield Diffie, Martin Helman

Future: Oded Regev? Jon Katz?