Key Exchange With Matrices and Lattices

October 14, 2019
DH and RSA Rely on Number Theory

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 (BQP).
5. If Quantum Computers (QC) ever become a reality then 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. Recent results by Google show that QC can do some things that classical computers cannot; however, this is a long way from factoring being easy.
3. I do not work in QC or Crypto; I have no special insights.
4. QC is worth studying for the insight it gives into both quantum and computing.
5. 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. Recent results by Google show that QC can do some things that classical computers cannot; however, this is a long way from factoring being easy.
3. I do not work in QC or Crypto; I have no special insights.
4. QC is worth studying for the insight it gives into both quantum and computing.
5. 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.
Assumes that Quantum Computing is Real. Consequences:

1. Factoring and DL are now easy. So can’t use DH or RSA.
2. Reductions.
   ▶  Standard Crypto: If BLAH is crackable then problem X is now easy. Easy means R (Poly time allowing coin flips and small prob of error). Used for a problem X we think is not in R.
   ▶  Post-Quantum Crypto: If BLAH is crackable then problem X is now easy. Easy means BQP (Quantum P). Used for a problem X we think is not in BQP.
True There is a Cryptosystem based on linear algebra that is post-quantum.
Whats True and Whats Educational

**True** There is a Cryptosystem based on linear algebra that is post-quantum. It’s secure but complicated.
True There is a Cryptosystem based on linear algebra that is post-quantum. It’s secure but complicated.

Educational I came up with a similar one. It’s simple but insecure.
True There is a Cryptosystem based on linear algebra that is post-quantum. It’s secure but complicated.

Educational I came up with a similar one. It’s simple but insecure.

Plan: I will teach the simple and insecure system that captures some of the ideas of the one that are complicated by insecure.
True There is a Cryptosystem based on linear algebra that is post-quantum. It’s secure but complicated.

Educational I came up with a similar one. It’s simple but insecure.

Plan: I will teach the simple and insecure system that captures some of the ideas of the one that are complicated by insecure.

Will then discuss the secure-but-complicated systems.
Terminology

The secure-but-complicated cryptosystem is called Learning With Errors—Key Exchange. Due to Regev
Terminology

The secure-but-complicated cryptosystem is called Learning With Errors—Key Exchange. Due to Regev abbreviated

LWE-KE
The secure-but-complicated cryptosystem is called Learning With Errors—Key Exchange. Due to Regev abbreviated

LWE-KE

The insecure-but-simple cryptosystem is called Learning With Gasarch—Key Exchange
Terminology

The secure-but-complicated cryptosystem is called Learning With Errors—Key Exchange. Due to Regev abbreviated

LWE-KE

The insecure-but-simple cryptosystem is called Learning With Gasarch—Key Exchange abbreviated

LWG-KE
LWG-KE

October 14, 2019
LWG-KE. Two Security Parameters $L, S$

1. Alice generates rand prime $p$ of length $M$, rand $S \times S$ matrix $A$ over $\mathbb{Z}_p$.
2. Alice sends $(p, A, SOTE)$. All public.
3. Alice generates rand $\vec{y} \in \mathbb{Z}_p^L$. Sends $\vec{y}A$.
4. Bob geneate rand $\vec{x} \in \mathbb{Z}_p^L$, Sends $A\vec{x}$.
5. Alice computes $\vec{y}(A\vec{x}) = \vec{y}A\vec{x}$.
6. Bob computes $(\vec{y}A)\vec{x} = \vec{y}A\vec{x}$.
7. Alice and Bob have shared secret $\vec{y}A\vec{x}$.
**LWG-KE. Two Security Parameters** $L, S$

1. Alice generates rand prime $p$ of length $M$, rand $S \times S$ matrix $A$ over $\mathbb{Z}_p$.
2. Alice sends $(p, A, SOTE)$. All public.
3. Alice generates rand $\vec{y} \in \mathbb{Z}_p^L$. Sends $\vec{y}A$.
4. Bob geneate rand $\vec{x} \in \mathbb{Z}_p^L$, Sends $A\vec{x}$.
5. Alice computes $\vec{y}(A\vec{x}) = \vec{y}A\vec{x}$.
6. Bob computes $(\vec{y}A)\vec{x} = \vec{y}A\vec{x}$.
7. Alice and Bob have shared secret $\vec{y}A\vec{x}$

*Secure? On HW you will show that it is not secure.*
Small Vectors

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

View $\mathbb{Z}_p$ as

$$\left\{-\frac{p-1}{2}, \frac{-p-3}{2}, \ldots, -1, 0, 1, \ldots \frac{p-3}{2}, \frac{p-1}{2}\right\}$$

Still do math mod $p$, but now it looks more Gaussian.
**LGE-KE. Two Security Parameters** \( L, S \)

1. Alice generates rand prime \( p \) of length \( L \), rand \( S \times S \) matrix \( A \) over \( \mathbb{Z}_p \).
2. Alice generates rand small \( \vec{y} \in \mathbb{Z}_p^L \), rand small \( \vec{e}_y \in \mathbb{Z}_p^L \). Sends \( \vec{y}A + 2\vec{e}_y \).
3. Bob generates rand small \( \vec{x} \in \mathbb{Z}_p^L \), rand small \( \vec{e}_x \in \mathbb{Z}_p^L \). Sends \( A\vec{x} + 2\vec{e}_x \).
4. Alice computes \( a = \vec{y}(A\vec{x} + 2\vec{e}_x) = \vec{y}A\vec{x} + 2\vec{y} \cdot \vec{e}_x \).
5. Bob computes \( b = (\vec{y}A + 2\vec{e}_y)\vec{x} = \vec{y}A\vec{x} + 2\vec{x} \cdot \vec{e}_y \).
6. (This is not true, alas). Alice and Bob both take what the computed mod 2 to both get \( yA\vec{x} \) (mod 2). So they both share a bit.

What they actually do is more complicated. For one thing, they only agree on the bit with high probability.
LWE-KE. Hardness Assumption

**Definition**

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

1. Pick a random $\vec{v} \in \mathbb{Z}_p^L$ small
2. Pick a random $e \in \mathbb{Z}_p$, small (you do not get to see $e$)
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 $L$) number of operations.

**Known** If LWE-KE is crackable then LWE is easy.

**So** Need to have a reason why LWE is hard.

**Want** A Hard Problem SVP such that LWE easy implies SVP easy.
Shortest Vector Problem (SVP)

SVP Given a lattice, find the shortest Vector out of the origin.

![Diagram of a lattice with shortest vector](Picture by Sebastian Schmittner - Own work, CC BY-SA 4.0, https://commons.wikimédia.org/w/index.php?curid=44488873)

**Hardness** Known to be NP-hard under randomized reductions.
Shortest Vector Problem (SVP)

SVP Given a lattice, find the shortest Vector out of the origin.

(Picture by Sebastian Schmittner - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=44488873)

Hardness Known to be NP-hard under randomized reductions.

Want $\text{SVP} \leq \text{LWE} \leq \text{LWE-KE}$. True, but with a Caveat.
LWE-KE. Hardness Assumption

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

Known: If can crack LWE-KE then can solve LWE.
Known: If can solve LWE then can solve SVP problem.
Upshot: If can crack LWE-KE then can solve SVP problem.
Caveat: The sense of can solve is odd—next slides.
LWE-KE. Hardness Assumption – A Caveat

We claimed:

\[ \text{SVP} \leq \text{LWE} \leq \text{LWE-KE} \]
We claimed:

\[ \text{SVP} \leq \text{LWE} \leq \text{LWE-KE} \]

This is true. Sort of.
We claimed:

\[ \text{SVP} \leq \text{LWE} \leq \text{LWE-KE} \]

This is true. Sort of.
It uses Quantum Reductions.
We claimed:

\[
\text{SVP} \leq \text{LWE} \leq \text{LWE-KE}
\]

This is true. Sort of.
It uses Quantum Reductions.
Recent Result Can replace Quantum with Randomized, as of last week.
LWE-KE. Hardness Assumption – A Caveat

We claimed:

\[ \text{SVP} \leq \text{LWE} \leq \text{LWE-KE} \]

This is true. Sort of.
It uses Quantum Reductions.
Recent Result Can replace Quantum with Randomized, as of last week.
Not Quite Actually I just learned it last week, it was known for two years, but the NSA only released it last week.
We claimed:

\[ \text{SVP} \leq \text{LWE} \leq \text{LWE-KE} \]

This is true. Sort of.

It uses Quantum Reductions.

Recent Result Can replace Quantum with Randomized, as of last week.

Not Quite Actually I just learned it last week, it was known for two years, but the NSA only released it last week.

I’m kidding
1. QC: DH cracked, LWE-KE uncrackable if GAP-SVP hard.
2. ¬QC: DH looks save, LWE-KE is classically hard.
1. **QC**: DH cracked, LWE-KE uncrackable if GAP-SVP hard.
2. ¬**QC**: DH looks save, LWE-KE is classically hard.

This is why post-quantum crypto uses quantum. Post-quantum crypto assumes Quantum computers exist and are fast.

- Can’t use number theory assumptions like factoring hard. :-(
- Can use quantum reductions to prove hardness results. :-(
How Important Is Public Key?

October 14, 2019
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
3. Facebook privacy – just kidding, Facebook has no privacy.
4. Every financial institution in the world.
5. Military – though less is known about this.
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 –
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.
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.
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 (RSA)
4. 2012- Silvio Micali, Shaffi Goldwasser
5. 2015- Whitfield Diffie, Martin Helman