The post-quantum Algorithm New Hope

A NEW PRINCIPLE FOR CRYPTOGAPHIC KEY EXCHANGE

Dipl. Math. Xenia Bogomolec, 15.02.2017

Content

Motivation
Personal Inspiration

I was always fascinated by cryptography. When I read, that there is a new method for a cryptographic key exchange, in which the key is not transmitted explicitly anymore I thought:

I really want to understand this, it sounds like magic!!!

Even more was I delighted when I found out, that the underlying math is based on the subject in which I researched at the Leibniz University Hanover, discrete algebraic geometry.

Simple Example: 31313 = 173*181

From about 100 digits it becomes difficult. A 1024-bit integer has 309 digits in decimal representation, a 4096-bit number has even 1234.

Motivation

Basic Necessity

The security of conventional cryptographic procedures is based on the conjecture, that it is not possible to compute prime factorization of big integers within reasonable time.

Peter Wiliston Shor (* 14. August 1959 in New York), an american mathematician and computer scientist, developed an algorithm for quantum-computers which can solve this problem.

This algorithm will be able to compute prime factorization of big integers within a time short enough, such that conventional methods like RSA, Diffie-Hellmann and ECC will not hold anymore.

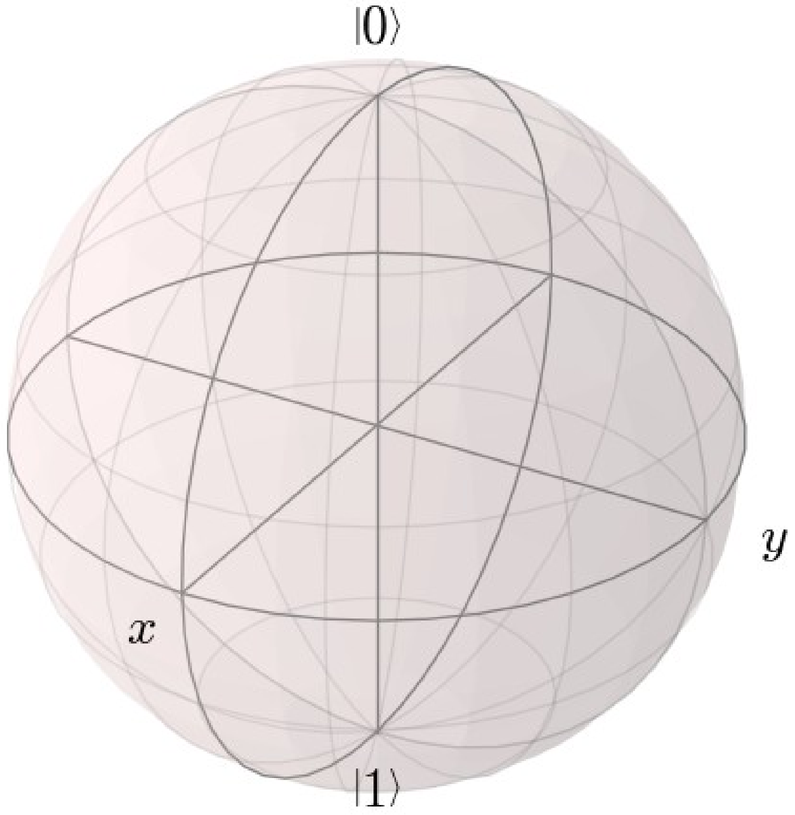
2001 Shor's algorithm was testet on a simple quantum computer with 7 qubits to factorize the integer 15.

Motivation
Basic Necessity

Binary Data (bits):

0 or 1

Quantum States (qubits):



Secret Keys

The secrecy of the keys to decrypt encrypted data is the security basis of any cryptographic procedure.

Different key-management for

- Symmetric procedures
- Asymmetric procedures
- Hybrid procedures

Secret Keys
Symmetric Procedures

Symmetric procedures use the same key for encryption and decryption of data.

Alice		Bob
Secret Key	=	Secret Key

→ Secret key exchange is necessary!

Secret Keys
Symmetric Procedures

Advantages:
- Efficient algorithms
- Simple key management

Disadvantages:
- Key exchange endangers encryption

So called one way functions allow mechanisms to mathematically hide the private key in the public key. They are easily computed in one direction, but hard to be inverted.

Secret Keys
Asymmetric Procedures

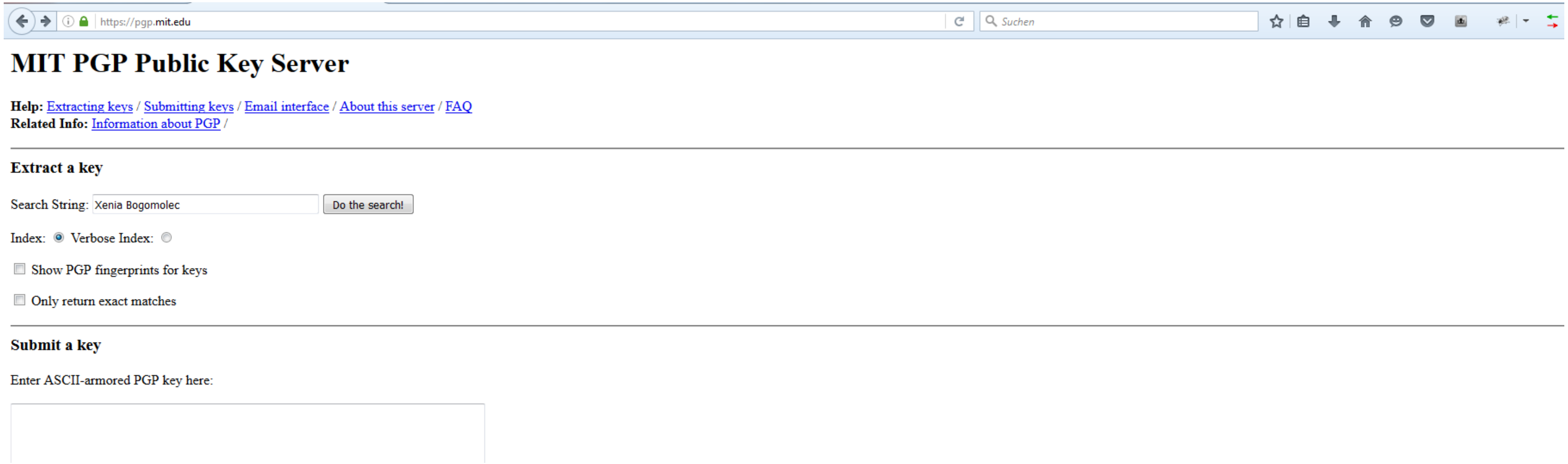
Asymmetric procedures use different keys for encryption and decryption of data. The data is encrypted with the public key and can only be decrypted with the private key.

Alice		Bob
Secret Key A	≠	Secret Key B
Public Key A	=	Public Key A
Public Key B	=	Public Key B

→ No secret key exchange necessary!

Secret Keys
Asymmetric Procedures

Advantages:
- High level security
- Key exchange does not endanger encryption!
  (There are even public key server)



Secret Keys
Asymmetric Procedures

Advantages:
- High level security
- Key exchange does not endanger encryption!
  (There are even public key server)

Disadvantages:
- 2 different keys per communication partner
- About 10 000 times slower than symmetric procedures
- Long keys

Secret Keys
Hybrid Procedures

Combination of symmetric and asymmetric procedure.

Symmetric:
Encryption of data with randomly generated *Session-Key*.

Asymmetric:
Session-Key is asymmetrically encrypted for transmission.

→ Asymmetrically encrypted key exchange!

Secret Keys
Hybrid Procedures

Alice		Bob
Session Key	=	Session Key
Secret Key A	≠	Secret Key B
Public Key A	=	Public Key A
Public Key B	=	Public Key B

→ Asymmetrically encrypted key exchange!

OpenSSL is a library with various cipher-suites for hybrid encryption.
The implemented standard is called TLS (Transport Layer Security) since 1999.

Secret Keys
Hybrid Procedures

Advantages:
- High level security
- Efficient algorithms for the big part of data
- Key exchange does not endanger encryption!

Disadvamtages:
- Complex key management
- Long keys

Key Exchange
Pre-quantum

Conventional procedures for key exchange:

- Diffie-Hellmann (1976)
- RSA (1977)
- Elliptic-Curve (1985)

Their security relies on the assumption, that the secret keys cannot be computed within reasonable time from public data.

Mathematically spoken, the security relies on the conjecture that prime factorization is hard to solve for big integers.

Key Exchange
Pre-quantum

Elliptic curves were introduced, because computations on them are much slower than on the ring of integers. Therefore shorter keys lead to the same level of security as longer keys with DH or RSA.

Mathematical fundamentals:

- Diffie-Hellmann:
  Discrete exponentiation $a = r^s$

- RSA:
  Product of large primes (key exchange) $n = p_1 \cdot p_2$
  Discrete exponentiation (crypto) $a = r^s$

- Elliptic-Curve:
  Multiplication of points on elliptic curves $a = r \cdot P$

Key Exchange
Pre-quantum

Inverse functions:

- Discrete logarithm
- Integer factorization
- Division of points on elliptic

Do not run within acceptable time on pre-quantum computers, even with the best known algorithms. (Not feasible in polynomial time complexity).

BUT:
With Shor‘s integer factorization algorithm quantum computers will be able to compute this invers functions within reasonable time.

Key Exchange
Post-quantum

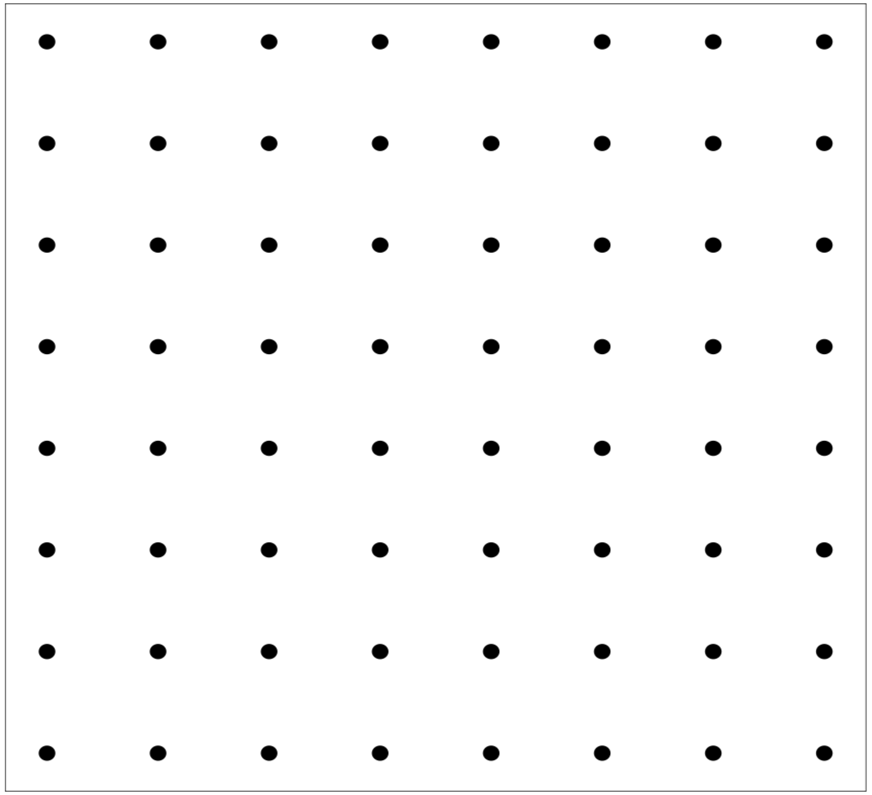
New mathematical fundamentals:
Lattice-based algorithms

What is a lattice?

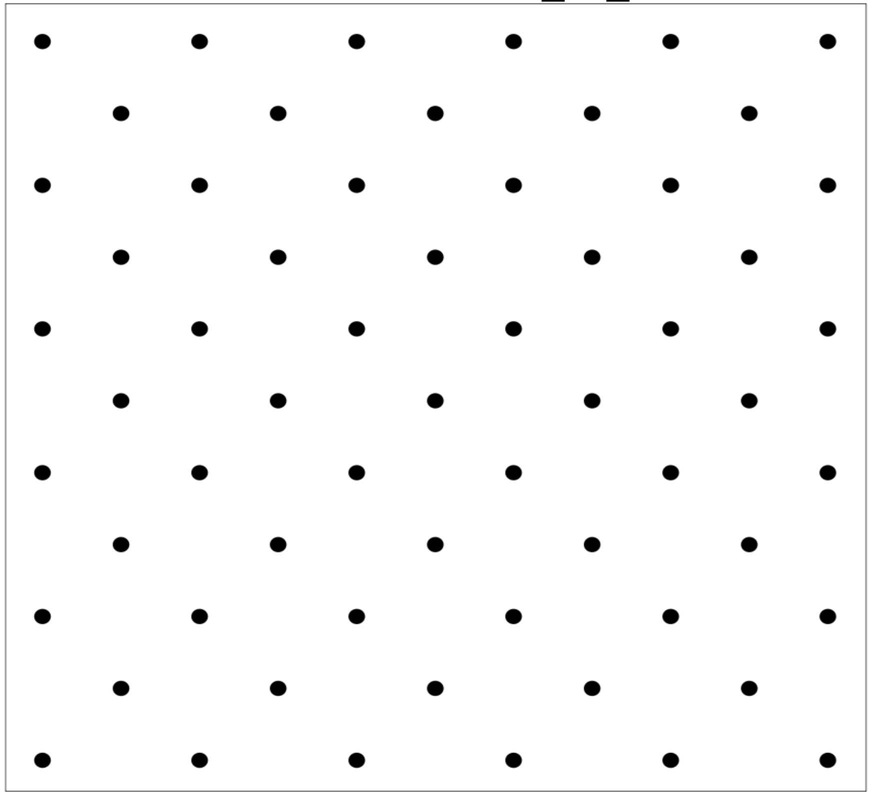
$$\Lambda = \left\{ \sum_{i=1}^{n} a_i v_i \;\middle|\; a_i \in \mathbb{Z} \right\}$$

Key Exchange
Post-quantum

$v_1 = (1,0), v_2 = (0,1)$



$v_1 = (1,0), v_2 = (\frac{1}{2}, \frac{1}{2})$



Examples for 2-dimensional lattices

Key Exchange

Post-quantum

Rooms of potential solutions for polynomial equations over the integers can be viewed as lattices.

Relation between integers and data transformation:
Each character or string can be interpreted as integer. *

Example for ascii encoded string:
Hallo = 48616c6c6f (hexadezimal) = 310‘872‘140‘911

* Many on rationals, reals or complex numbers easily invertible functions become one way functions on the integers.

Key Exchange

Pre- vs. post-quantum

The mightiness of the room of possible solutions stands for the possibilities an attacker has to try, if he cannot attack the algorithm or the involved systems. This is called Brute Force Attack.

Pre-quantum:

$$a = r^s$$
$$n = p_1 \cdot p_2$$
$$a = r \cdot P$$

→ Room of possible solutions: integers or points on EC (1-dimensional)

Post-quantum:

$$a(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 \cdot x + a_0$$
$$a_i \in Z, \quad i \in \{0, \ldots, n\}$$

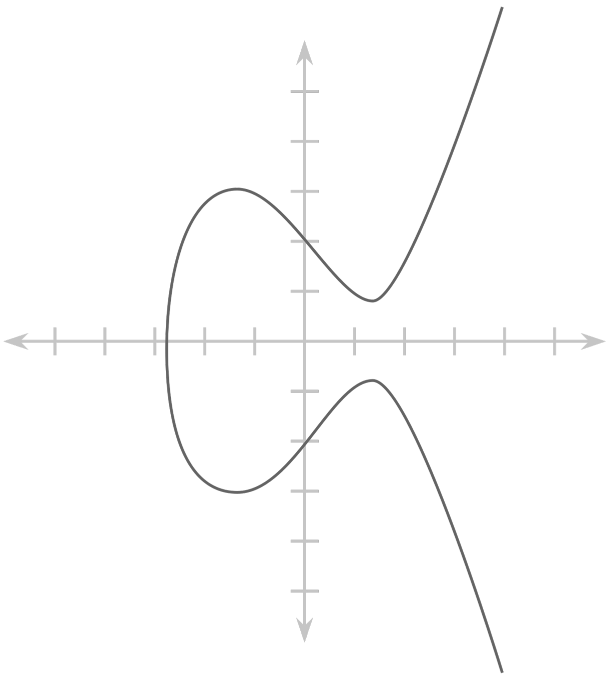
→ Room of possible solutions: 2-dimensional lattice

Key Exchange

Pre- vs. Post-quantum

Pre-quantum:

und



Post-quantum:

Key Exchange
Gitter

Conjecture as mathematical term:

1. Stronger than a simple assumption
2. Everything points to fact that it holds
3. But there is no proof of it

Hardness of worst-case lattice-problems is a conjecture

→ Fundamental of security for many post-quantum problems

New Hope

Ne explicit key exchange anymore.

New principle: Key Encapsulation Mechanism

→ Chris Peikerts (MIT) KEM with more efficient parameters

New Hope

Peikerts KEM

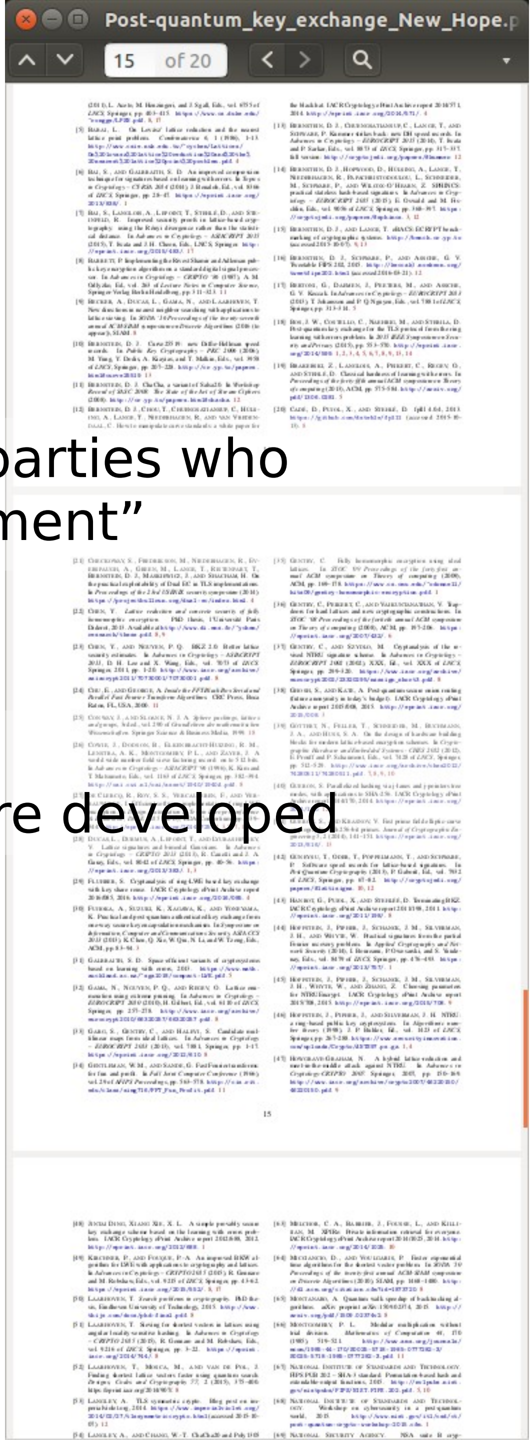
„ A simple, lowbandwidth reconciliation technique that allows two parties who 'approximately agree' on a secret value to reach exact agreement"

Simple?

… compared to the methods, from which KEM and New Hope are developed

Mathematical background: Ring Learning with Errors

Excerpt of references from the New Hope paper →



New Hope

Peikerts KEM

„ A simple, lowbandwidth reconciliation technique that allows two parties who ‘approximately agree’ on a secret value to reach exact agreement”

Ring Learning with Errors

$$
\begin{array}{ccccccccc}
14s_1 & + & 15s_2 & + & 5s_3 & + & 2s_4 & \approx & 8 \; mod \; 17 \\
13s_1 & + & 14s_2 & + & 14s_3 & + & 6s_4 & \approx & 16 \; mod \; 17 \\
6s_1 & + & 10s_2 & + & 13s_3 & + & 1s_4 & \approx & 3 \; mod \; 17 \\
 & & & & & & & \vdots & \\
6s_1 & + & 7s_2 & + & 16s_3 & + & 2s_4 & \approx & 3 \; mod \; 17
\end{array}
$$

New Hope

Peikerts KEM in Key Exchange

Preconditions: polynomial of degree 1024 with coefficients from $\{0, \dots, 12288\}$

Alice	Bob	Alice
$p_a(x) = a(x) \cdot s_a(x) + e_a(x)$	$p_b(x) = a(x) \cdot s_b(x) + e_b(x)$	
Sends $p_a(x)$ and $a(x)$ to Bob	$v(x) = p_a(x) \cdot s_b(x) + e_b(x)$ KEM → - Session Key k - Reconciliation string c	
	Sends $p_b(x)$ and c to Alice	$w(x) = p_b(x) \cdot s_a(x) + e_a(x)$ Reconciliation string c KEM → - Session Key k

New Hope

KEM vs. Diffie-Hellmann

Alice	Bob	Alice
$p_a(x) = a(x) \cdot s_a(x) + e_a(x)$	$p_b(x) = a(x) \cdot s_b(x) + e_b(x)$	
Sends $p_a(x)$ and $a(x)$ to Bob	$v(x) = p_a(x) \cdot s_b(x) + e_b(x)$ *KEM* → - Session Key k - Reconciliation string c	
	Sends $p_b(x)$ and c to Alice	$w(x) = p_b(x) \cdot s_a(x) + e_a(x)$ Reconciliation string c → - Session Key k

Alice	Bob	Alice
a, g, p $A = g^a \bmod p$	b $B = g^b \bmod p$	
Sends A, g, p to Bob	$K = A^b \bmod p = g^{ab} \bmod p$	
	Sends B to Alice	$K = B^a \bmod p = g^{ba} \bmod p$

New Hope

Possibilities that have to be tried in a brute force attack

$$s(x)\ New\ Hope: 1024^{12289} = 3{,}76 \cdot 10^{36993}$$

$$p_1, p_2\ RSA: \leq 3{,}13 \cdot 10^{1232}$$

New Hope
Man in the Middle

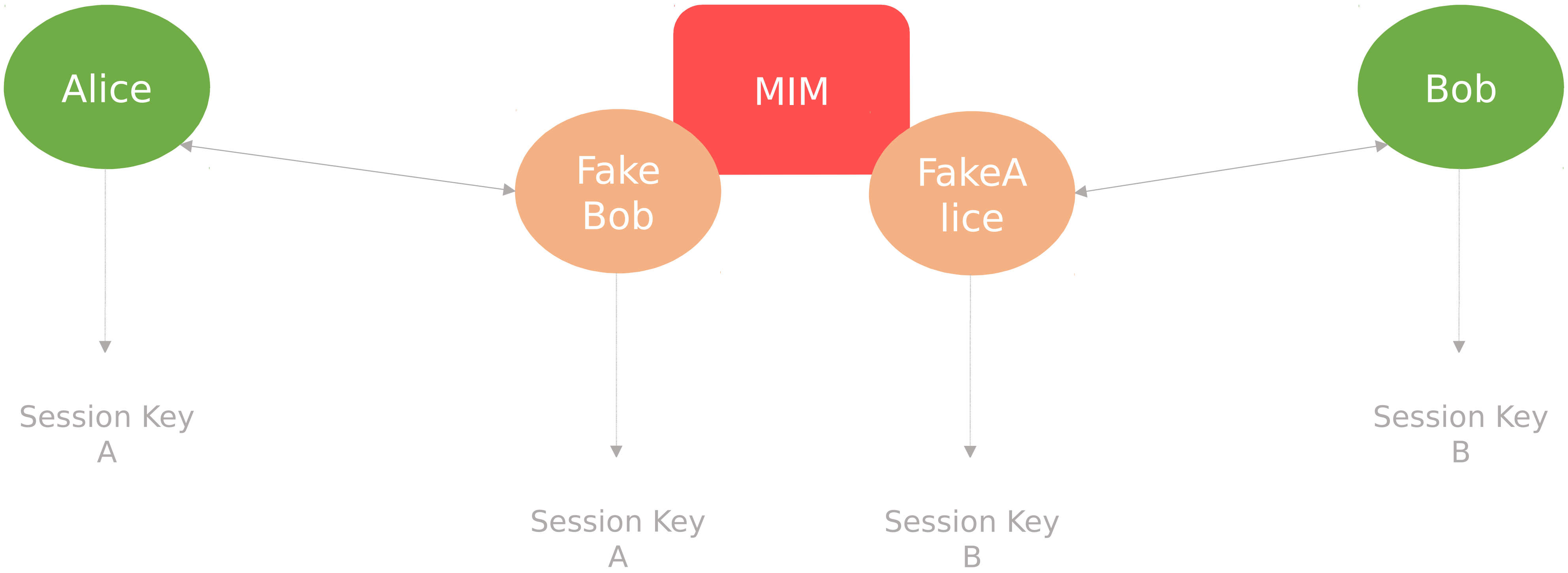
Man in the Middle is possible if

- KEM not authentificated
  (✓ in current implementation)
- Polynom $a(x)$ known
  (Optional, but in New Hope not!)
- Computation parameters known $(1024, 12289)$
  (✓public, actually this parameters make the KEM usable)

BUT: Google embeds New Hope in an ECC-procedure

New Hope

Man in the Middle



Googles Test

New Hope was testet on a few connections between chrome and the google servers 2016.

A test phase of about 2 years should challenge the crypto community to attack the New Hope algorithm.

To protect the users, New Hope was embedded in a ECC-procedure, the algorithm is called CECPQ1.

The test phase was interrupted after a few months.

Googles Test
End of the test phase

Two scientists published a paper with an algorithmic solution for the 'closest vector problem in lattices' on 21.11.2016. It was withdrawn 24.11.2016 due to an error.

On 28.11.2016 Adam Langley announced the early ending of the New Hope test with the following reasoning:

1) The published and withdrawn paper was considered an achieved goal.
2) On very slow connections, the New Hope key exchange could have a to strong impact.
3) They assume that there only exist very simple quantum computers.
4) The integration of New Hope in TLS 1.3 could be too complex.
   (TLS 1.3 consists of one less round trip than TLS 1.2)

Appendix

Random Values

The choice of the polynomials $s(x), e(x)$ in $p(x) = a(x) \cdot s(x) + e(x)$ succeeds from a so called central binomial distribution.
$a(x)$ is generated for each session with a SHAKE-128 from a 256-bit seed.
The security diminution compared to from a noise generator * produced values is as small, that it can be negelcted.

RNG** can be replaced by a PRNG***!

*   Physical source for random values
** Random Number Generator
*** Pseudo Random Number Generator

(Each software-based generator only produces pseudo-random values.)

Appendix
Error Probability

The probability of Alice computing a different session-Key than Bob berechnet is smaller than

$$2^{-60} = 8{,}67 \cdot 10^{-19}$$

That means that in less than a trillion connections, one side would receive non-sense data. If that happens, a new session is initiated.

Appendix
Message Lengths

Sender	Nachricht
Alice	1824 Bytes
Bob	2048 Bytes

A number from the set $\{0,\ldots,12289\}$ can be represented by two bytes (FFFF = 65535).
A polynomial of degree 1024 could be represented by $2 \cdot 1024 = 2048$ Bytes.
This length can be reduced by a number theoretic transformation.

Appendix

Numbers New Hope

Possibilities for $s(x), e(x)$ in $p(x) = a(x) \cdot s(x) + e(x)$: $1024^{12289} = 3{,}76 \cdot 10^{36993}$

New Hopes parameters for computations:

- *1024* is the dimension of the ring in which the computations happen (upper bound for degree of a polynomial)
- *12289* is the modulus (coefficients of the polynomial are from the set *{0,..., 12288}*)

In modular Computations a number/polynomial can be the product of larger numbers/polynomials.

Example: $5 \cdot 6 = 30 \equiv 2 \, mod \, 7$

Appendix
Numbers RSA

Possibilities for $p_{1,} p_2$ in a 4096-bit public RSA key $n = p_1 \cdot p_2$:

Bytes	Hexadezimal	Formel dezimal
1	FF	$2^8 - 1$
2	FFFF	$2^{2 \cdot 8} - 1$
512	512 X FF	$2^{512 \cdot 8} - 1$

If we find p_1, p_2 is found too. 0 and 1 are no primes. If $p_1 = 2$ then n is even, what we would see immediately. All other even numbers and all numbers ending with 5 are no primes, such that there remain

$$\left(2^{512 \cdot 8} - 2\right)/2 - \left(2^{512 \cdot 8}/5\right) = \left(2^{512 \cdot 8 - 1} - 1\right) - \left(2^{512 \cdot 8}/5\right) = 3{,}13 \cdot 10^{1232}$$

numbers to test.

Appendix
Sophisticated RSA-Hacks

The primes $p_{1,} p_2$ can be testet prime number lists.

The currently largest known prime is $2^{74'207'2810} - 1$.

If the prime factors are close to $\sqrt{n}$, they can be found with probalistic methods within seconds.