Motivations for the use of Semigroups in (post-quantum) cryptography

Ambroise Grau

University of York

April 15, 2020

Cryptography in a nutshell

Cryptography is the study of secure communication protocol, where security relates to three principal concepts:

1. confidentiality: only the relevant parties have access to the information
2. data integrity: the information transmitted has not been altered during the transport
3. authentication: the receiver can verify the signature of the sender in the communication

Remark: If the communication channel is secure (e.g. a diplomatic case), then the information can be sent into plaintext and all the requirements will be met by the carrier. On an insecure channel (e.g. internet) we often have to encrypt the information to meet some of the requirements.

Classical vs quantum cryptography

- Classical: all operations can be done on a classical computer using bits
- Quantum: some operations necessitate the use of a quantum computer and qubits

Is it a big difference?
Yes!... but quantum computers are not necessarily more powerful than classical ones, they are simply tackling a problem differently and thus can be more efficient at *some* tasks (such as factorising and computing discrete logarithm).

Post-quantum cryptography

Definition

Cryptographic algorithms running on classical computers that resist attacks using classical and quantum computers.

Why?

- Quantum computers are currently being built and are becoming more and more powerful.
- A network of global quantum communication is currently unfeasible.
- We need to be able to stay secure using our current computers.

These are the main reasons of the strong interest in PQC, leading to the call to proposal from NIST with the goal to roll out new standards by 2024.

What do we want in order to build a good cryptosystem?

1. Decide which problem we want to address (there is not a unique solution doing everything right);
2. Use a "hard problem" that cannot be broken by a computer;
3. Establish a protocol;
4. Show that this is sound and seems secure (or at least try);
5. Find examples that fits in there.

What do we want in order to build a good cryptosystem?

1. Decide which problem we want to address (there is not a unique solution doing everything right):
   if we want to achieve a full encryption, we will need all communicating parties to have a key, but if we are building a signature scheme, only the author is involved in the set up, and a register needs to hold its final signature for verifications.
2. Use a "hard problem" that cannot be broken by a computer;
3. Establish a protocol;
4. Show that this is sound and seems secure (or at least try);
5. Find examples that fits in there.

What do we want in order to build a good cryptosystem?

1. Decide which problem we want to address;
2. Use a "hard problem" that cannot be broken by a computer: there are here two different meanings of the word "cannot" to assert security[1]:
   - Computational security: "cannot" means "computationally infeasible even given access to a quantum computer."
   - Conditional security: "cannot" is to be understood as secure "assuming some mathematical problem is hard on a quantum computer."

   For this compare the hardness (NP complete, NEXP...) of the studied problem with other well-known ones.
3. Establish a protocol;
4. Show that this is sound and seems secure (or at least try);
5. Find examples that fits in there.

[1] Albrecht M., *The road to Post-Quantum Cryptography*, workshop on Cyber Security, York, 2019.

What do we want in order to build a good cryptosystem?

1. Decide which problem we want to address;
2. Use a "hard problem" that cannot be broken by a computer;
3. Establish a protocol:
   design how keys are created and exchanged (core of the research), as well as decide when they should be discarded.
4. Show that this is sound and seems secure (or at least try):
   firstly it should be possible to run the protocol in reasonable time, and then the efficiency of variants of known attacks should be assessed on our protocol to verify it is not already broken.
5. Find examples that fits in there:
   sometimes easier to do abstract mathematics, but we need to go back to the practical use here and find semigroups satisfying all requirements that can be implemented in memory and that would allow the protocol to run with them as platform.

What do we want in order to build a good cryptosystem?

1. Decide which problem we want to address;
2. Use a "hard problem" that cannot be broken by a computer;
3. Establish a protocol
4. Show that this is sound and seems secure (or at least try)
5. Find examples that fits in there

Only after completing these steps we will have a "proto-secure" protocol that is worth publishing to be challenged by others (where "proto-secure" has to be understood in the sense that we expect it to be secure, but it is likely that some attacks against it could be built).

Well-known problems: DHP, DLP, SAP

Let S be a semigroup.

Definition (Diffie-Hellman problem)

Given elements s, s^a and s^b in S where $a, b \in \mathbb{N}$ are secret, recover the value s^{ab}.

Definition (Discrete Logarithm Problem)

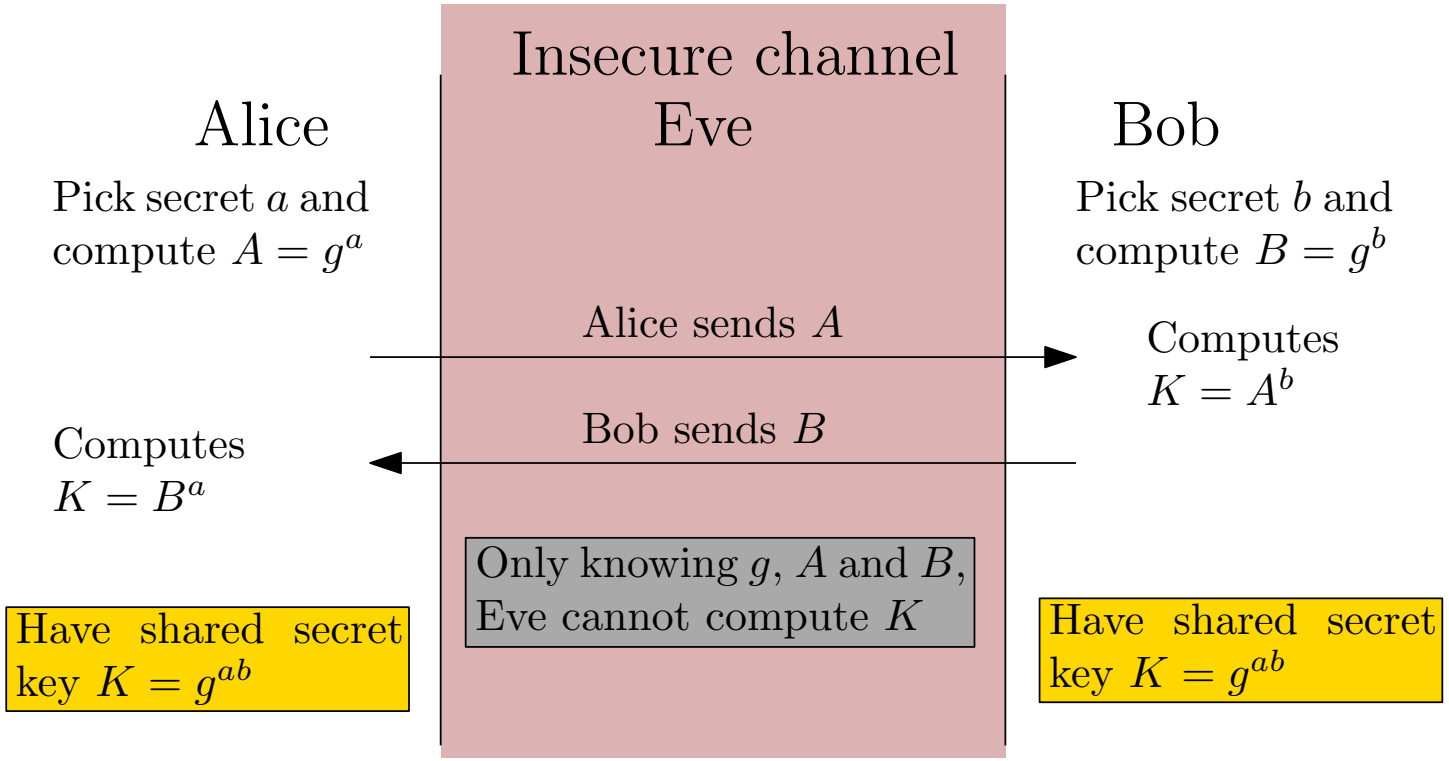
Given $s \in S$ and $t \in \{s^k \mid k \in \mathbb{N}\}$, find a value $x \in \mathbb{N}$ such that $s^x = t$.

Definition (Semigroup Action Problem)

Given S acting on a set T and elements $t \in T$ and $y \in St$, find $x \in S$ such that $xt = y$.

The Diffie-Hellman Key Exchange protocol

Alice and Bob want to communicate through an insecure channel. In order to create a shared secret key, they agree on a group G and a generator $g \in G$ and run the following protocol:



Semi-direct product of semigroups

In a serie of articles (2013-2016), Kahrobaei et al. defined a protocol under semidirect product of semigroups:

Definition

Let S be a semigroup, $\phi \in \mathrm{End}(S)$ and $T = \langle \phi \rangle$, then in $S \rtimes T$ the multiplication is as follows:

$$(s_1, \phi^n) \cdot (s_2, \phi^m) = (s_2 \phi^n \cdot s_1, \phi^{m+n}).$$

The DHKE is then worked by agreeing on a public element $g \in S$ and then Alice and Bob both choose an secret exponent (say a and b) and compute $(g, \phi)^e$ with e their exponent, and they will end up sharing the key $(g, \phi)^{a+b}$.

However, Roman'kov in 2015 showed that if the semigroup S is not well-chosen, some linear algebra attacks can be performed (using some cyclic behaviour of the action), and thus this is not necessarily resistant.

Using a semigroup presentation I

Kropholler et al. decided to analyse the class of semigroups of the form:

$$S = S(p, r, q, s) = \langle a, b \mid a^p = b^r, a^q = b^s \rangle,$$

where $p, q, r, s \in \mathbb{N}$, and $gcd(p, q) = 1$, and they proved the following:

- there exists a complete rewriting system in terms of normal forms;
- relies on the hardness of the DLP in the semigroup;
- the set $T(p, r, q, s)$ of normal forms $a^u b^v$ (with some restrictions on the order of p, r, q, s) gives an ideal of S isomorphic to the cyclic group of order $ps - qr$;
- the DLP and SAP are easy on T (this is doable simply by Euclid's algorithm and reduction to normal form) making the DHKE insecure.

Using a semigroup presentation II

However unsuccessful, their attempt to use a semigroup with a given presentation was to satisfy the conditions Shpilrain, Ushakov and Zapata gave in 2006 for a semigroup S to be considered as a platform:

1. There should be an efficiently computable normal form for each element of S.
2. It should be computationally easy to perform semigroup operations on normal forms.
3. There should be an efficient algorithm to disguise elements of a chosen subset T of S.

Is the DLP hard enough?

Quick answer: Not in general!

Childs and Ivanyos (2013) proved that in any finite semigroup, there is an efficient quantum algorithm to compute the DLP. Independently, Banin and Tsaban showed that in a periodic semigroup, solving the DLP is equivalent to solving it in cyclic subgroups of this semigroup. Since there is a quantum algorithm (due to Shor) to compute discrete logarithms in groups, any element with finite order is not safe to use when the underlying problem of the protocol is the DLP.

What about elements with infinite orders? We don't know exactly...

Other approaches

In 2003 Kim and Moon postulated that the use of the class semigroups of imaginary quadratic non-maximal orders could give rise to some encryption schemes and devised a key-exchange system. However, in 2004 Jacobson proved that some structural properties of such semigroups rendered their cryptosystem insecure and than any other built in the same way would not provide more security than using groups.

Ustimenko and Klisowski (2019) wrote a paper titled "On Noncommutative Cryptography and homomorphism of stable cubical multivariate transformation groups of infinite dimensional" to propose a problem they call the Cremona Semigroup Word Decomposition (CSWD) supposedly resistant to quantum algorithms. But no one has yet been able to verify this as the maths involved are quite complicated and very specific.

Why I believe the study of semigroups can be important for post-quantum cryptography I

Many problems thoroughly studied for cryptography in group theory can be shifted to semigroups in a more complicated way such as:

- Conjugacy search/decision problems:
  - in groups: $g \sim h \iff \exists k \in G : g = khk^{-1}$;
  - in semigroups: for $a \in S \setminus \{0\}$, let $\mathbb{P}(0) = \{0\}$, $\mathbb{P}(a) = \left\{u \in S \mid (ma)u \neq 0, \forall ma \in S^1 a \setminus \{0\}\right\}$ and then $a \sim_c b \iff \exists u \in \mathbb{P}^1(a), v \in \mathbb{P}^1(b)$: $au = ub$ and $bv = va$.
- Nilpotency:
  - in groups: look at derived series $G^{(n)} = [G^{(n-1)}, G]$, and then G is nilpotent if $G^{(n)}$ is trivial for some n;
  - in semigroups: define the words q_i in the variables $x, y, z_1, z_2, \cdots \in S$ recursively by $q_1(x, y) = xy$ and $q_{i+1}(x, y, z_1, z_2, \ldots, z_i) = q_i(x, y, z_1, z_2, \ldots, z_{i-1}) z_i q_i(y, x, z_1, z_2, \ldots, z_{i-1})$; and call S nilpotent if $q_c(x, y, z_1, z_2, \ldots, z_{c-1}) = q_c(y, x, z_1, z_2, \ldots, z_{c-1})$ for some c.
- Word problems, hidden subset problems, etc.

Why I believe the study of semigroups can be important for post-quantum cryptography II

Other problems only exist in the context of semigroups, for example: given $s \in S$ is s in the subsemigroup generated by $E(S)$?
Some ideas are also coming from usual semigroup theory: James Renshaw worked on E-dense semigroups and completely regular semigroups and proposed a cryptosystem built on the latter that is more robust than current ones built on groups. However this is not post-quantum as it relies on the hardness of a modified version of the DLP.

My personal point of view: semigroups provides a very good setting for the creation of signature scheme, and can give rise to very good encryption system in some regular or inverse subclasses (whenever inverses are needed), but ideas can only come from doing semigroups and encountering hard problems that can be carried over to computer science rather than trying the other way around.