A passive Dolev-Yao intruder that reads *XOR*

Mohammad Torabi Dashti
CWI, Amsterdam

Verification of cryptographic protocols

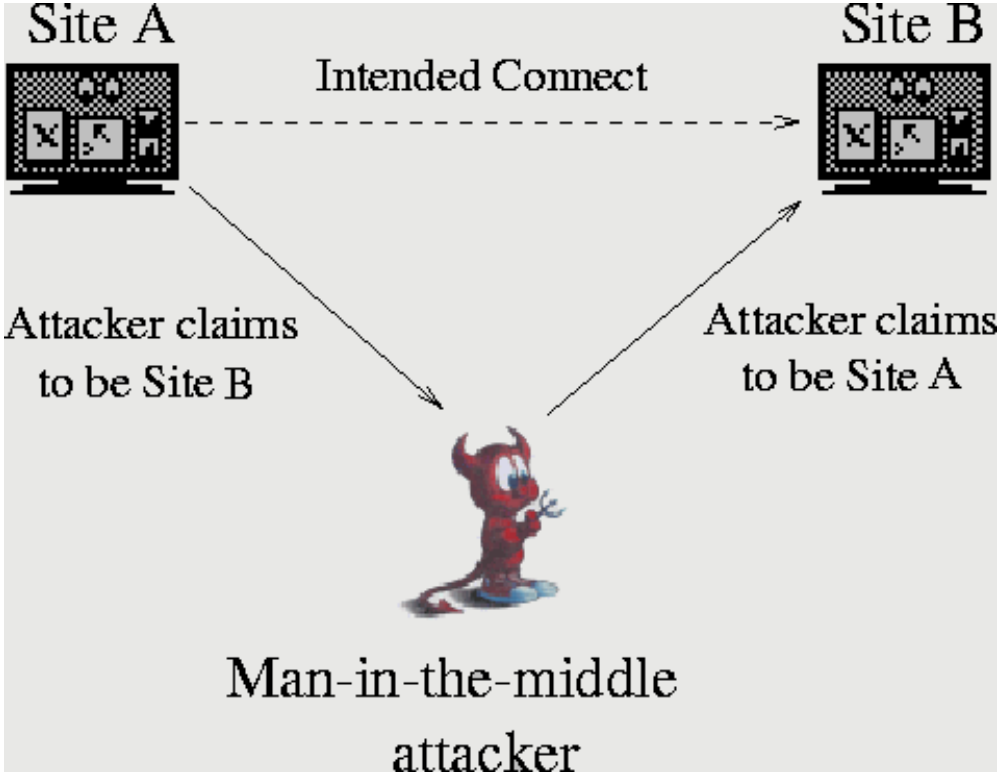
- "Formal" approaches (*FM)*: Looking at concurrency issues, abstracting away from (non-trivial) cryptographic properties.
- Computational approaches (*CC)*: Computational complexity view.

FM: A quick review - I

- Formal methods has traditionally been used to address concurrency issues in distributed systems and algorithms.
- Dolev and Yao [1984] presented a formal model of intruder to verify security protocols, basically abstracting away from cryptographic components, addressing concurrency-related attacks.

FM: A quick review – II

- An example of attacks detected in DY model:



FM: A quick review – III

+ It is (usually) possible to automatize the verification process.
- The analysis has a high level of abstraction, so some attacks could be missed.

CC: A quick review

- Attacker is a Turing Machine, with Polynomially-bounded resources.
- Usually the analysis is hard to automatize, but once done, the protocol is secure modulo mathematics.
- How to combine *FM* and *CC*?
  - Add *CC*-aspects to *FM*
  - Study in which situations *FM* results hold in *CC* framework [Abadi, Rogaway 2000]

DY + *XOR*: I

- DY intruder does not have the ability to see the underlying mathematical properties of operators used in cryptography.
- Amongst the most prevalent mathematical operators is *XOR* $\oplus$.
- Let's see how we can add *XOR* to DY.

DY + *XOR*: II

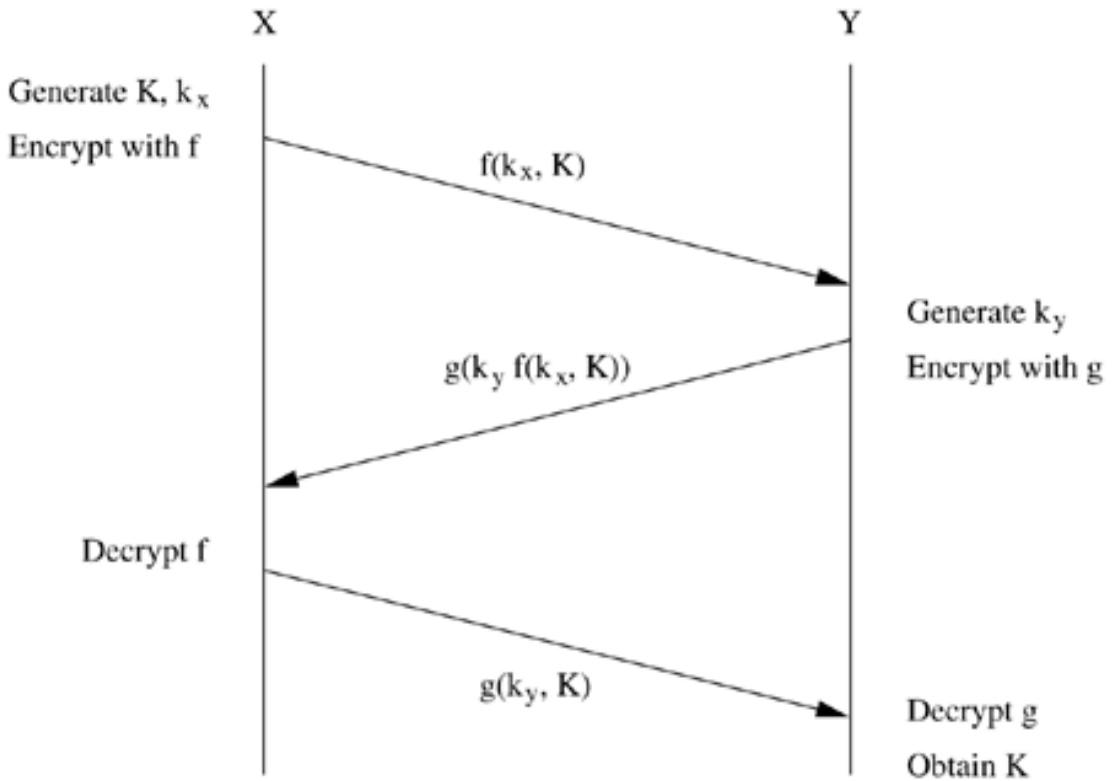
- Passive intruder has to be able to decompose messages and add to its knowledge using *XOR* properties.
- The set of messages that can be generated by deliberate application of *XOR* on set X is decidable, because:

$$n > 0.X^{\oplus n} \subseteq X \cup X^{\oplus 2} \cdots \cup X^{\oplus |X|}$$

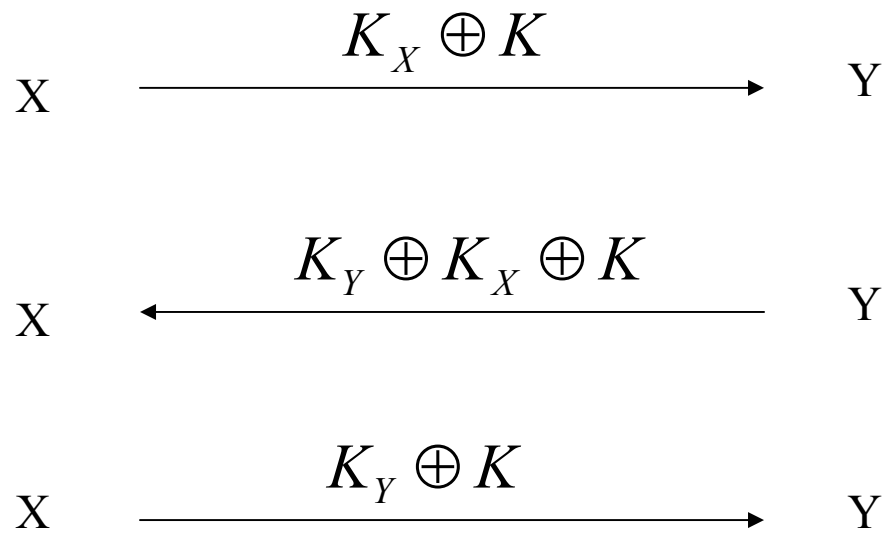
DY + *XOR*: III

- The previous algorithm was implemented in μ-CRL process algebraic language and used to find a well-known attack for Shamir's three-pass protocol. The attack is hidden to the eyes of the DY intruder model.

Shamir's three-pass protocol - I



Shamir's three-pass protocol - II



The next step?

- This method can not be directly extended to Diffie-Hellman exponentiation, for instance, which is another important mathematical operation used in crypto-protocols.
- Arithmetic is not decidable, so what can be done?
- Looking back at decidability results for DY intruder model…

Decidability results on DY

- The security of crypto-protocols is un-decidable. Sources of un-decidablity are:
  - Unbounded number of sessions.
  - Unbounded (number and) length of messages.
- The insecurity of crypto-protocols is NP-complete once the number of sessions in bounded [Rusinowitch, Turuani 2001].

What it means?

- The essence of the theorem: if there is an attack, its size is polynomially bounded. Then, it is easy: guess a protocol run and check if it constitutes an attack.
- Efficient algorithm to check it: constraint solving by Shmatikov and Millen 2001: Fix a protocol run, use a constraint solver to see if an attack is feasible. What is missed? Automatic check of concurrency-based attacks.

Being ambitious!

- We like to have the whole story in a process algebra, generating interleaving runs and checking if they are attacks.
- Side note: What we lack now is type-flaw attacks.
- If we have a constraint solver integrated with our process algebraic tools, there is a hope to benefit routines which has already been developed to decide about decidable fragments of arithmetic.

End,
and then …