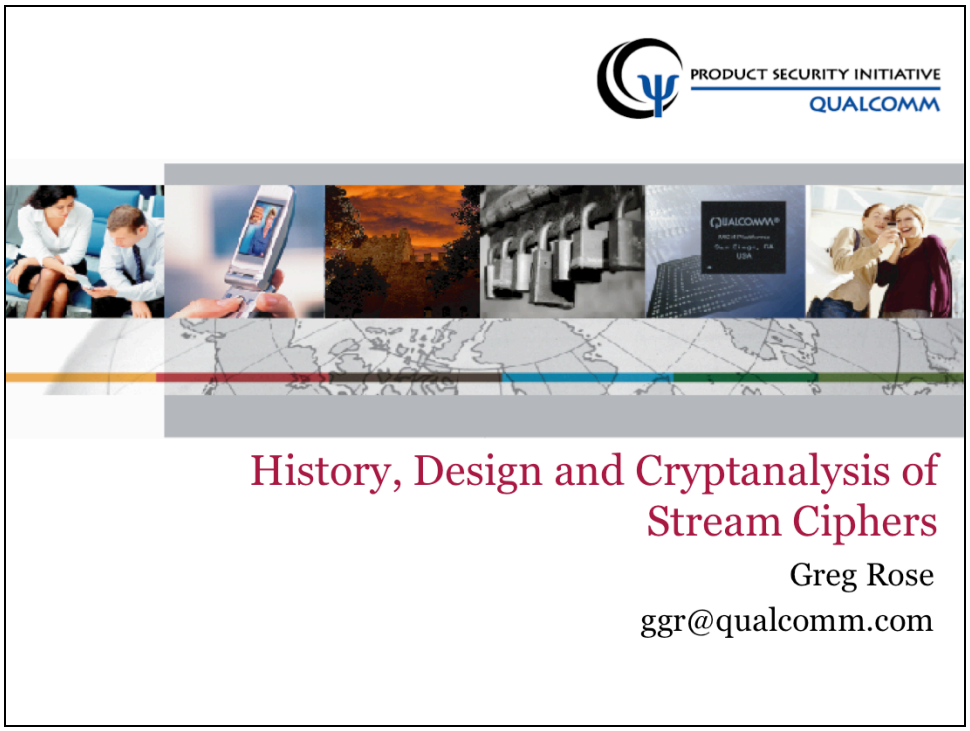
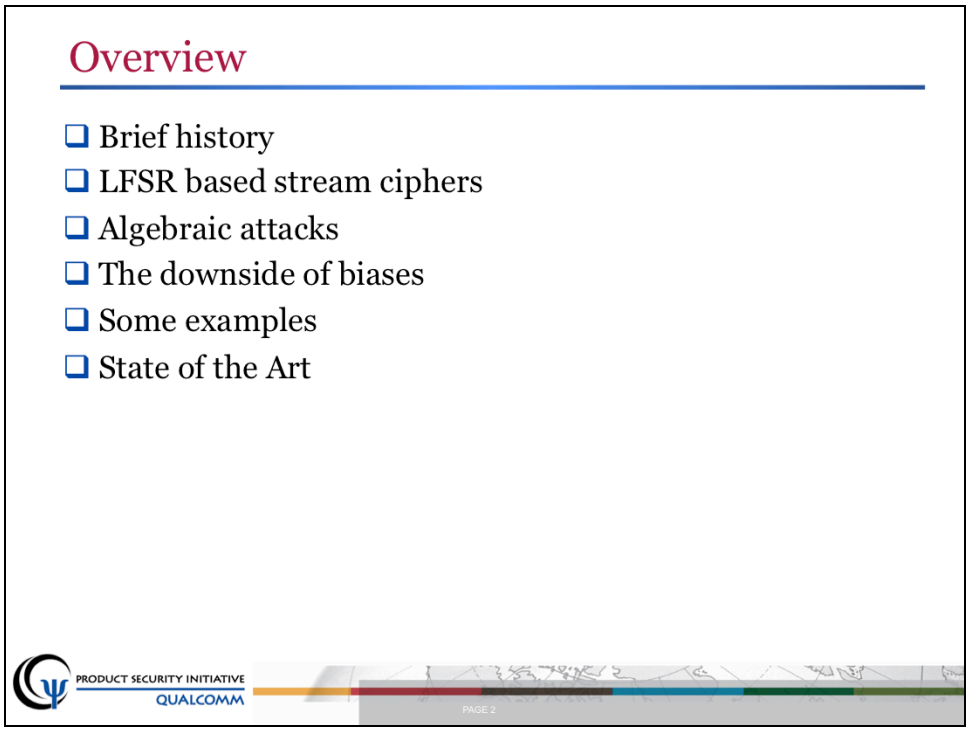


Greg Rose, QUALCOMM Australia

Greg Rose graduated from the University of New South Wales with a B.Sc. (honours) in computer science and was awarded the University Medal in 1977. A member of the Board of Directors of the USENIX Association, he served as program chair of the 1996 USENIX Security Symposium. As Senior VP for product security at QUALCOMM, he focuses on cryptographic security and authentication for wireless communications. He has written a number of public tools using cryptography, and he holds generic cryptographic export licenses for two countries.



I feel it is always enlightening to see some of the history behind a subject.

There is much more to cryptography than algorithms, such as key management, protocols, and so on.This tutorial does not attempt to address these other things.

What is a Stream Cipher?

- Unclear up to the mid 1800s.
- Characterized by operations to encipher small units
  - characters, bytes, down to bits
- Steps along the way to stream cipher:
  - Vigenère (polyalphabetic) cipher
    - (Alberti 1467, Bellaso 1553)
    - "le chiffre **indéchiffrable**"
  - Autokey cipher
  - Running key cipher
- Also plaintext/ciphertext feedback
  - usually insecure, not addressed in this talk

PRODUCT SECURITY INITIATIVE QUALCOMM

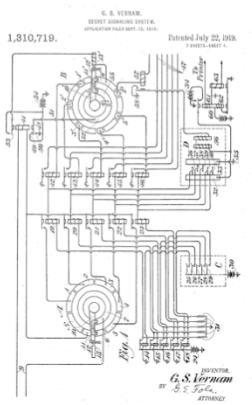
PAGE 3

Image of Blaise de Vigenère courtesy of Wikipedia.

Giovan Battista Bellaso in his 1553 book *La cifra del. Sig. Giovan Battista Bellaso*

The One Time Pad

- Miller in 1882 applied random offsets to codebook encryptions
  - requirements for randomness and no re-use spelled out
- Vernam (1917) modifies teletype
  - XOR with relays, bit level encryption
  - Tape in loop; mechanical Vigenère
- Mauborgne (1920?)
  - Rediscovers(?) requirement for randomness and no re-use
- Claude Shannon during WW2
  - founder of information theory
  - proves security of true OTP



PRODUCT SECURITY INITIATIVE QUALCOMM

PAGE 4

In 1882, a California banker named Frank Miller published Telegraphic Code to Insure Privacy and Secrecy in the Transmission of Telegrams. (See *Frank Miller: Inventor of the One-Time Pad* Steven M. Bellovin, Department of Computer Science, Columbia University)

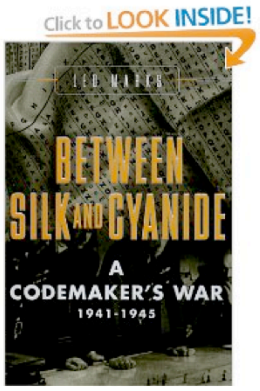
Gilbert Sandford Vernam at Bell Labs

Capt. Joseph Mauborgne, US Army Signal Corps

Claude Shannon at Bell Labs, work done during WW2, published 1948, founded information theory

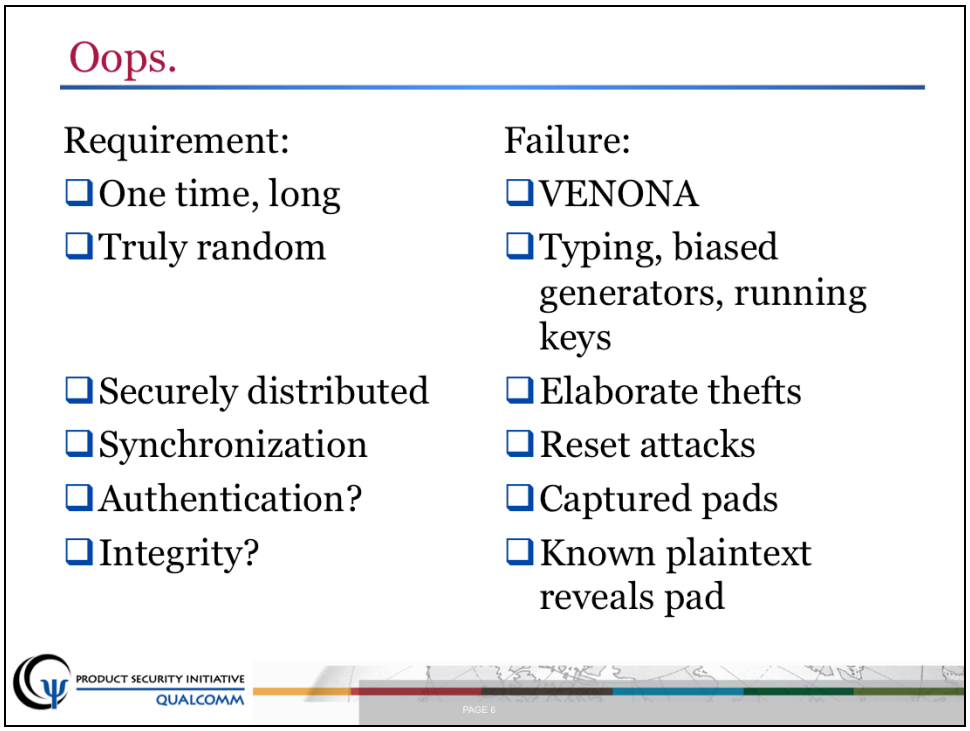
The One Time Pad!!! OMG!!!

- Start with lots of truly random numbers
  - lots and lots and lots…
  - truly truly random…
- Somehow make two (only two) copies
- Securely distribute the copies
- Carefully synchronize everything
- Never re-use any of them. Preferably destroy them. Rice paper, Yum!



- Shameless plug for Leo Marks' book.

Image shamelessly stolen from Amazon



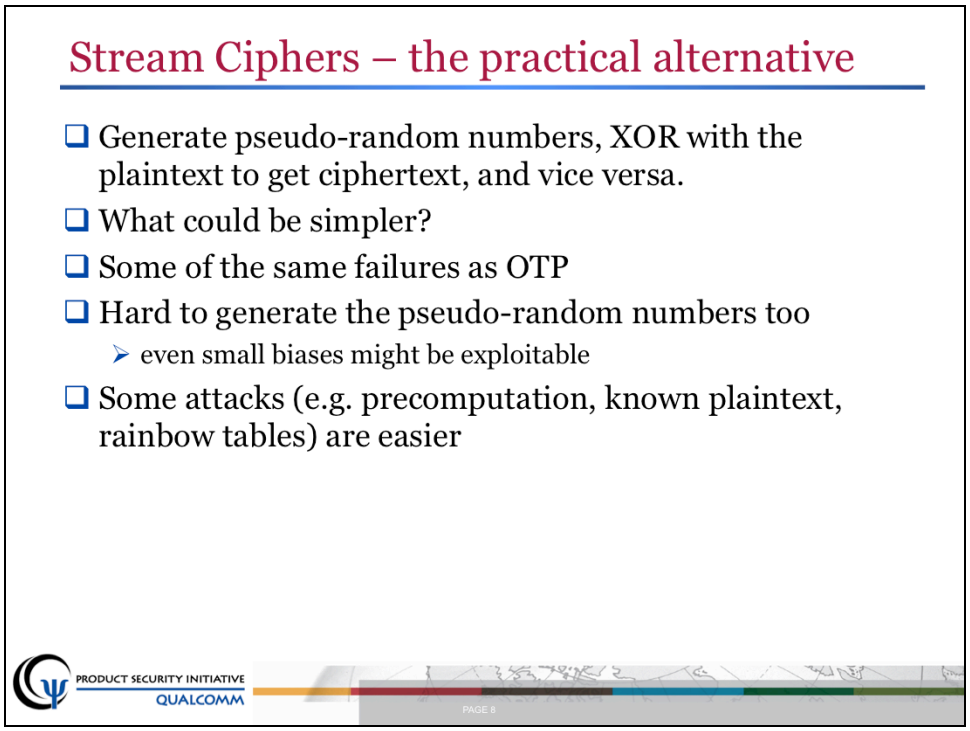
VENONA: USSR accidentally re-used some of their one-time pads. NSA ran huge correlations on archived data, turned them into running-key ciphers, decrypted.

Most amazing OTP – SIGSALY

PRODUCT SECURITY INITIATIVE
QUALCOMM

PAGE 7

National Cryptologic Museum, just outside NSA headquarters in Fort Meade, Maryland. (Converted motel!) Used to encode and encrypt voice transmissions between Roosevelt and Churchill during WW2. Random pad distributed on phonograph records. 50 tons, 30kW.



If you reuse a key, the same stream of output is generated. If you take two encrypted samples and XOR them together, the stream cipher cancels out, and you have two chunks of data XORed together. This is called a running key cipher, and is surprisingly easy to break using statistical properties of the plaintexts. (It doesn't matter *how* you combine the stream, there's always (by definition) a way to make it cancel out.)

Because you get nothing for free, you always must use stream ciphers together with other things; random numbers to enhance keys, some way to agree on keys, MACs or hashes to authenticate and bind data, some way to guarantee synchronisation.

Algebraic attacks

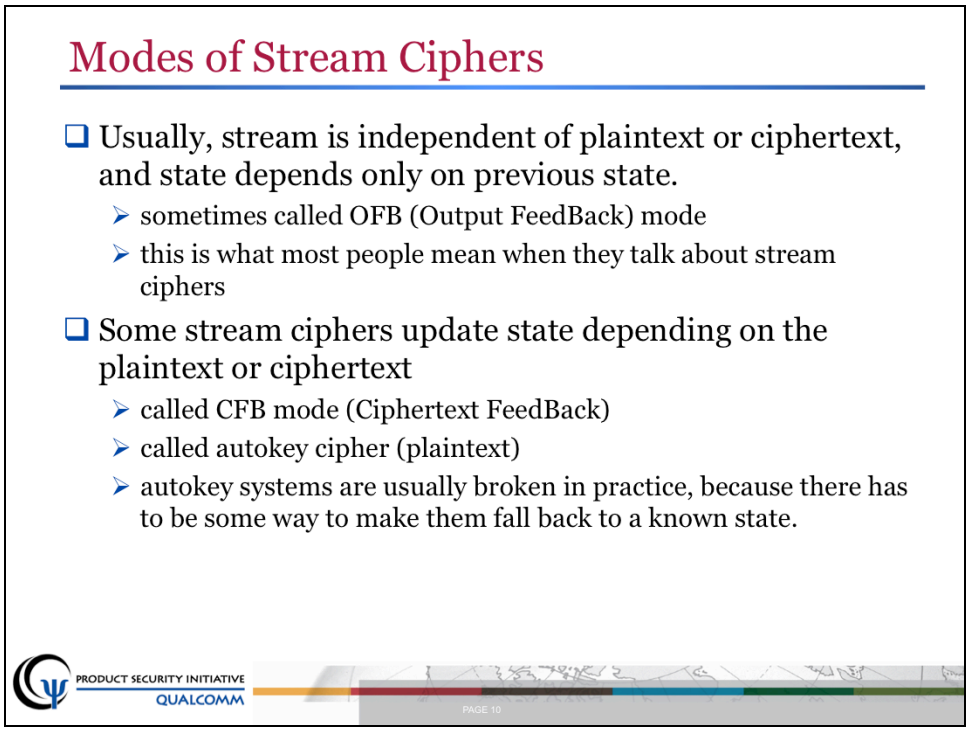
- Sometimes called "linearization" attacks
- Any cipher can be described as a multivariate high-degree polynomial
- Usually these are intractable
- … but if the degree isn't too high…
  - Treat each monomial as a new variable
  - Solve equations as a linear system
- Attacks on
  - Serpent (S-boxes too small)
  - Rijndael (debatable) (algebraic structure in S-box)
  - Many, many stream ciphers.
  - A5/2 (low algebraic degree, only binomials)

PRODUCT SECURITY INITIATIVE QUALCOMM

PAGE 9

See many papers by Courtois and others

Painting of Karl Friedrich Gauss by Christian Albrect Jensen



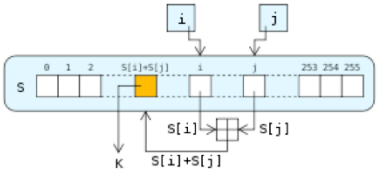
Note that in the latter case, you are not so much attacking the cipher itself, as the way in which it must be used in practice.

CFB mode stream ciphers must be very strong indeed, because they leave the door wide open for chosen-plaintext attacks.

The intelligence community uses different terminology in this area than is common in the open literature, and it can get quite confusing. What I call *key* they call *cryptovariable*, and what I call *stream* or *keystream* they call *key*.

RC4™

- Another of Ron Rivest's brilliantly simple designs
- Used in many standards
- Outputs a stream byte-at-a-time
- Variable length key up to 256 bytes
- 258 bytes of state
  - byte permutation (state) table *S*
  - counter *i*, bouncy index *j*
- very fast
- Biases in output
  - in particular, digraphs (0,0) are too common



PRODUCT SECURITY INITIATIVE QUALCOMM

PAGE 11

Image from wikipedia. Biases courtesy of Fluhrer and McGrew, then many others.

When i == 0 (known to the attacker), and first output is 0, probability of second being zero is 1/128, exactly twice what it should be.

The Server in a Cave Attack

- Capture a server somewhere with a ~160GB disk full of encrypted documents
  - RC4 encryption, multiple keys
- Assume documents use Unicode 16-bit glyphs
- Gather statistics of 16-bit glyphs assuming the (0,0) digraph frequencies
- The captor can determine the language of the documents!

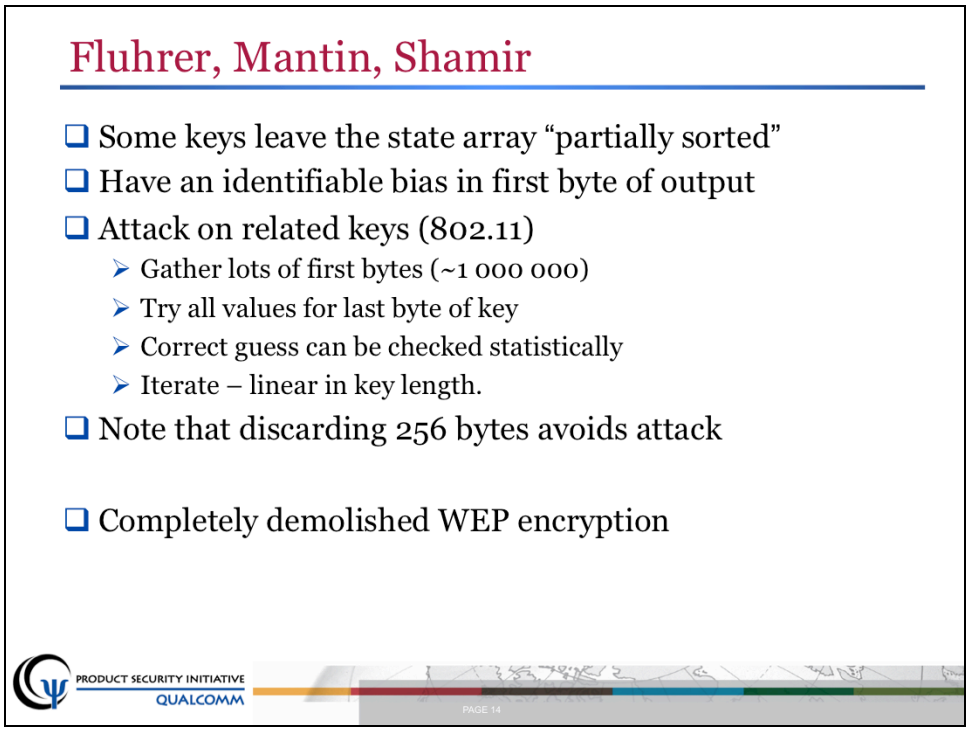
PRODUCT SECURITY INITIATIVE
QUALCOMM

PAGE 12

Image is of the Wikileaks server a former cold war bunker is located in the Pionen White Mountain in Sweden. See http://bitshare.tumblr.com/post/2383169988/the-wikileaks-server-cave

A note on biases and distinguishing attacks

- Note that the preceding attack is independent of the key length
  - Also independent of the keys!
- Distinguishing attacks work on block ciphers too
  - 64-bit block in counter mode is distinguishable in 2^{35} bytes!
  - whether single or triple DES
  - rarely held to the same standards
- Distinguishing attacks depend on the amount of data encrypted by the key holder
  - offline attacks are meaningless
  - attacker can't force the victim to do more work



This slide is a gross oversimplification of a very mathematical attack, which is nevertheless very simple and elegant.

The paper is to appear at Selected Areas of Cryptography (Toronto, August 15-17, 2001).

Folks at AT&T Labs (Rubin, Ioannides, ???) who already had 802.11 sniffer stuff lying around, implemented the attack in a few days and found that it takes about 15 minutes of intercepted traffic to recover the key (regardless of the key length).

Note that the attack even works on *unknown* plaintext, so long as there is an identifiable bias (eg. English text, IP address), it just requires more input.

But now...

- Before we can explore shift register based stream ciphers or public-key algorithms, we need some...

(gulp)

... mathematics

Recurrence relations

- Any sequence where n^{th} element is defined by an equation involving previous elements
- Example: Fibonacci numbers:
  - $F_k = F_{k-1} + F_{k-2}$
  - Initial state: $F_0 = F_1 = 1$
  - 1, 1, 2, 3, 5, 8, 13, 21, ...
- *Order* of the recurrence relation is number of terms right side "goes back"
- Relation to polynomial equation:
  - $x^2 - x - 1 = 0$

This is just the simplest example.

The polynomial form is called *the characteristic polynomial* of the recurrence relation.

General form of a k-th order *linear* recurrence relation:

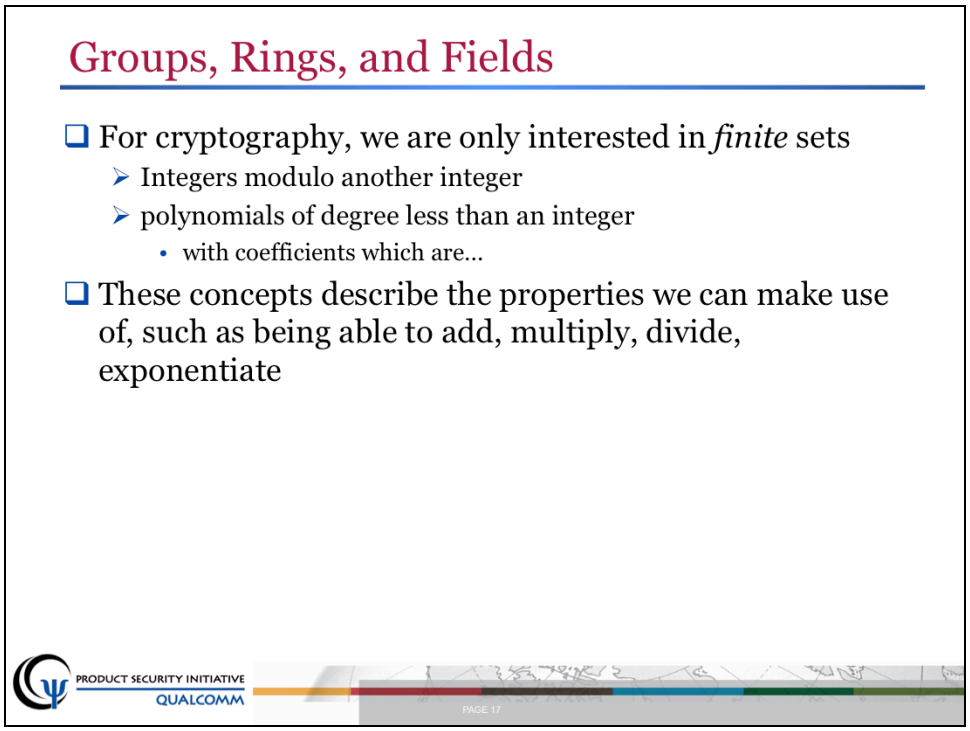
$$x_n = c_0x_{n-1} + c_1x_{n-2} + \ldots + c_{k-1}x_{n-k}$$

Characteristic polynomial:

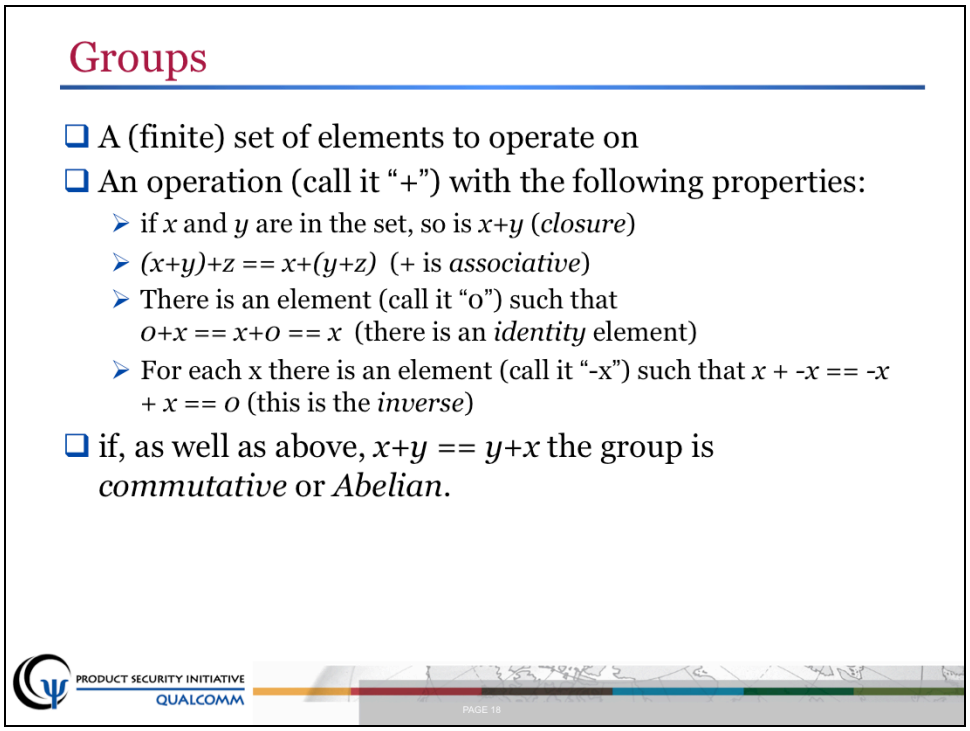
$$X^n - c_0x^{n-1} - c_1x^{n-2} \ldots - c_{k-1}$$

Note that when the coefficients are over a field of characteristic 2, "+" is the same as "-", so people often write the characteristic polynomial with "+" signs and this gets confusing. More later.

The *order* of the recurrence relation is the same as the *degree* (greatest exponent) of the polynomial.



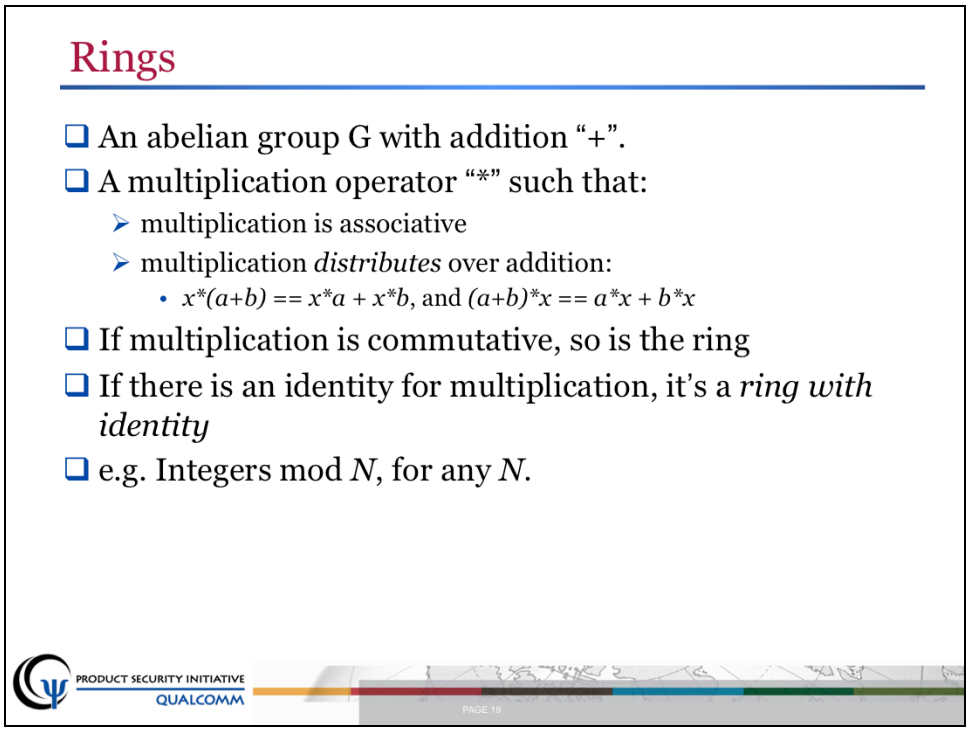
Of course there are infinite sets too, like the *ring of integers* or the *field of real numbers*. We just generally don't get to use truly infinite things.



For those with applicable upbringing, the group properties can be remembered as CAIN (Closure, Associative, Identity, iNverse) and ABEL.

When referring to a group, you get to choose whether you talk about it *additively* as above, or using *multiplicative* notation, where you think of the operator as multiplication, the identity as "1", and the inverse as $^1/_x$ or x^{-1}. This doesn't matter much until you start thinking about rings.

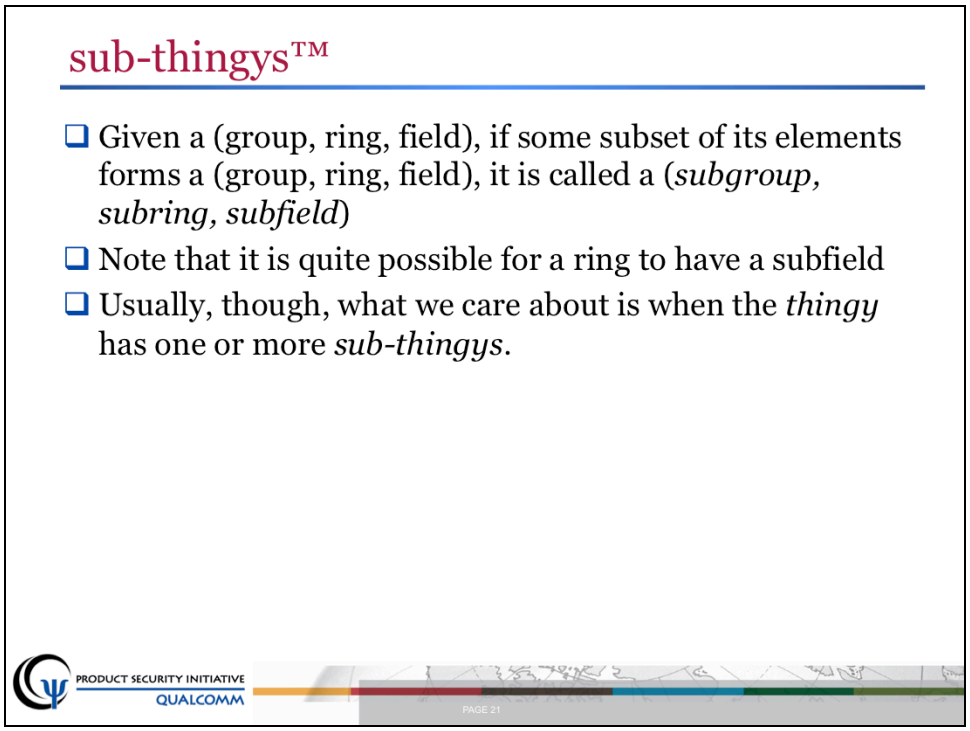
Matrix multiplication is a group most people are familiar with that is *not* abelian.



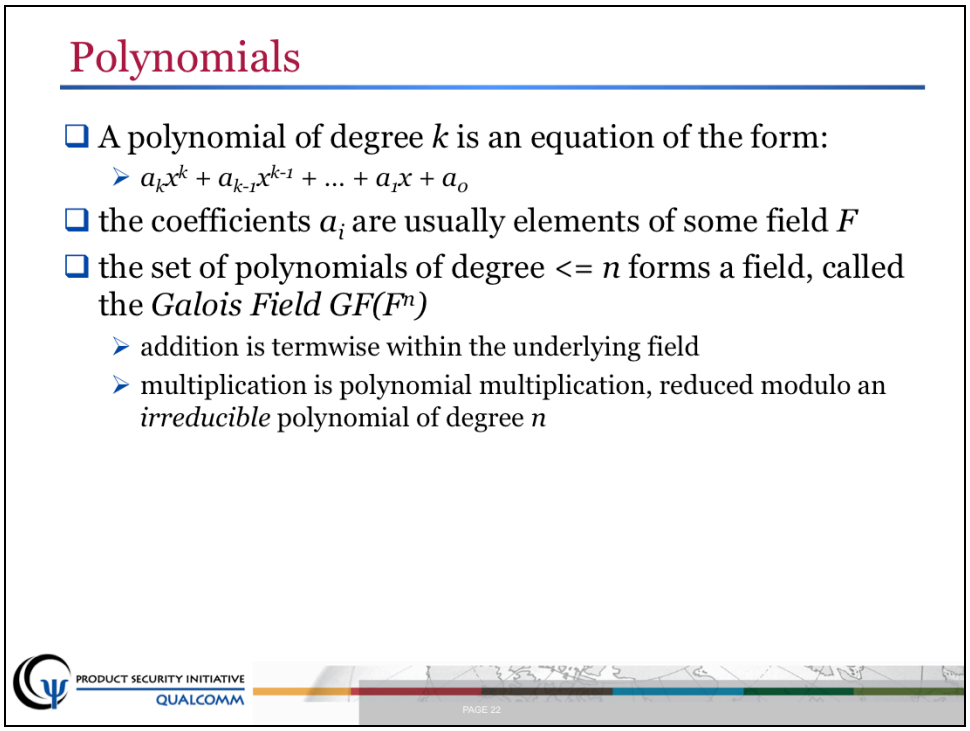
Yes, there are rings which don't have an identity element, for example the set of *even* integers… (*1* is not an element of the set, but all the other rules apply). Note that we're talking about the *multiplicative* identity here… there must be an additive identity because G is a group.

Fields

- A field is a set F
  - If it's an abelian group for addition
  - and a ring
  - and is an abelian group for multiplication if you ignore 0 (which can't have a multiplicative inverse…)
- The set F^* (F leaving out 0) is itself a group, and is called the *multiplicative group* of the field F
  - note: *not* a subgroup -- the operation is different
- e.g. Integers mod P, where P is prime



"*sub-thingys*" is not a very technical term.



What is *x*? Who cares? Think of it more as a placeholder for sorting out the arithmetic, and not a variable or something to be solved for.

Polynomials generally (when not limited in size, or when the field itself is not finite) form a *ring with identity.*

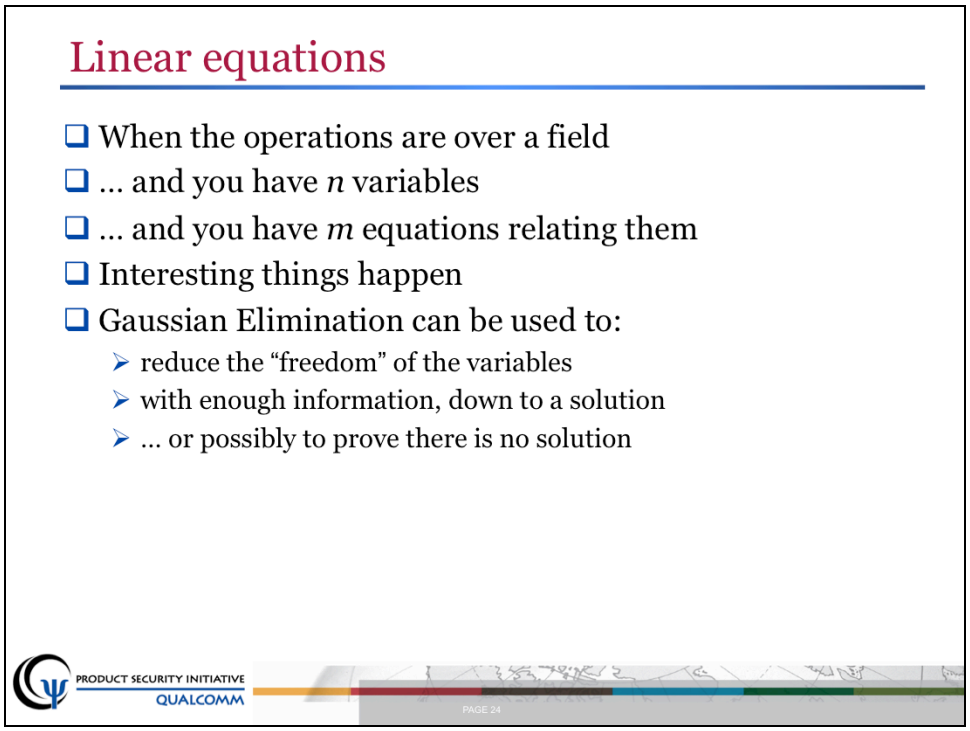
Evariste Galois died in a duel at the age of about 21… a terrible waste.

Enough of this for now? Good. We'll be back.

An important field: F_2

- The integers modulo 2 form a (small) field.
- Addition modulo 2 is *XOR*
- Multiplication modulo 2 is *AND*

- Integers modulo 2^n ($n > 1$) do *not* form a field
  - even numbers have no multiplicative inverses
- Polynomials of degree *(n-1)* over F_2 *do* form a field
  - $GF(2^n)$.



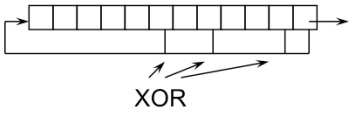
Many aspects of cryptanalysis boil down to describing things so that they act as fields, then gathering enough information to apply Gaussian Elimination. Sean Murphy's result about Twofish mentioned earlier is a sort of example.

If you have *n* independent equations in *n* variables, you have a solution. By *independent* we mean that the equations don't duplicate (or contradict) information you already had. The exact definition of this is too complicated for here.

Karl Friedrich Gauss is who this is named after.

Linear Feedback Shift Registers

- A bunch of bits, obeying a recurrence relation.
- Easily implemented in hardware
- This is called the *Fibonacci* configuration



- A pain in software
- Characteristic polynomial above is
- $C(x) = x^{12} + x^6 + x^4 + x + 1$

The polynomial shown is *primitive* (I know, 'cause Schneier told me so…). Remember, those "+"s are really "-"s, but that's the same thing in a field of characteristic 2.

LFSRs in software

- There's a software equivalent, called the *Galois configuration*:
- *if (r & (1 << 12))*
  - *r = (r << 1) ^ 0x53;*
- *else*
  - *r <<= 1;*
- Note the difference in shift direction
- Sequence of bits generated is the same, but the *state* of the register at a given point is different

Galois configuration

- If the state of the register is considered as a polynomial
- ... the shift left is "multiply by x"
- and the XOR is "reduce modulo $C(x)$".

LFSR update

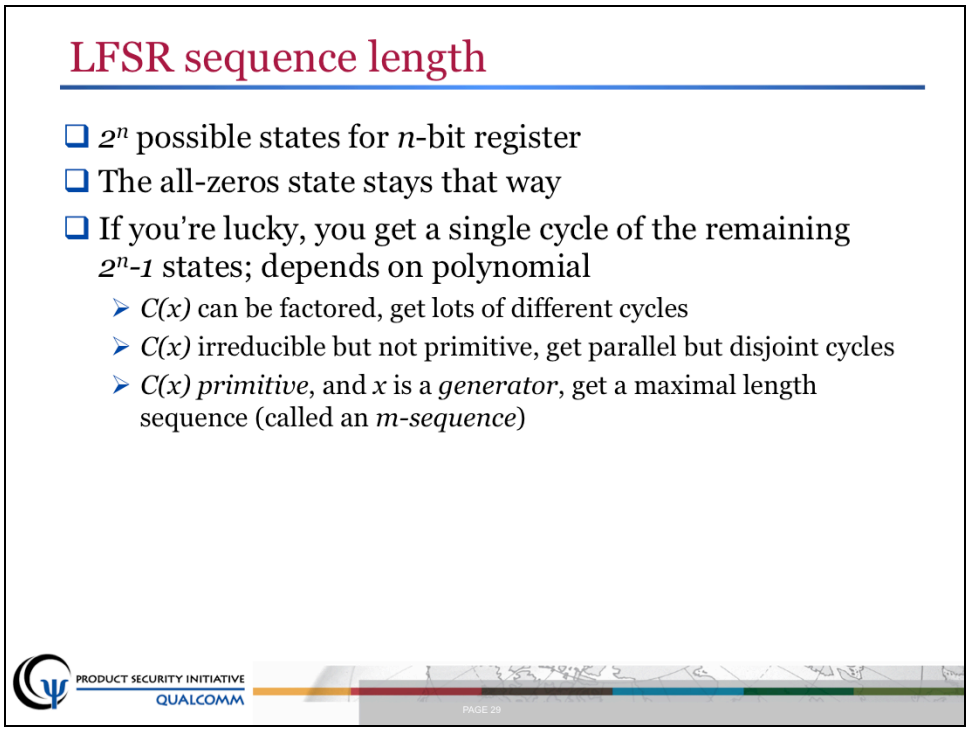
- Updating the LFSR (either style) can be viewed as a matrix multiplication

$$\bar{s}_{n+1} = \bar{s}_n A$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

There are many different ways of writing this, pre- or post-multiplication, row or column vector, but the point is it can be done fairly simply, and this equivalence makes all sorts of results easy to figure out.

Note that the first column of the matrix (in this form) is the bits of the recurrence relation.

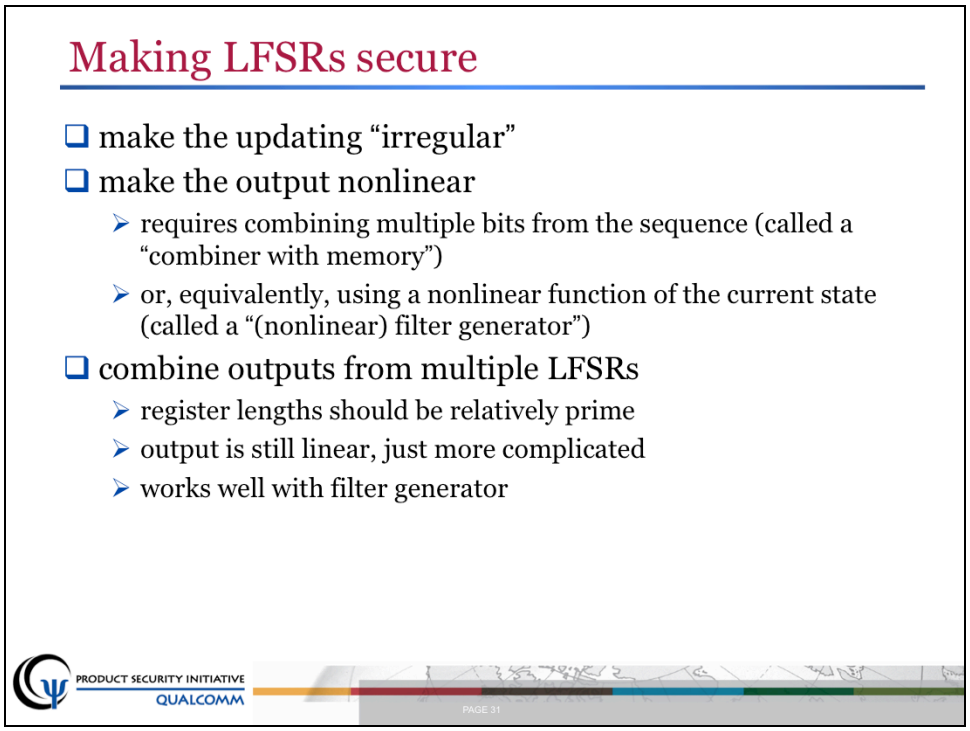


There are different ways of defining "primitive" and "generator", and in fact the definitions are related in interesting ways. These concepts will come up later, in a context where it is easier to explain them, under Public-key Cryptography.

For the time being, I'll just note that a perfectly good definition of the terms is "if you have a maximal length sequence generated by a shift register, then the characteristic polynomial must have been primitive".

About LFSRs

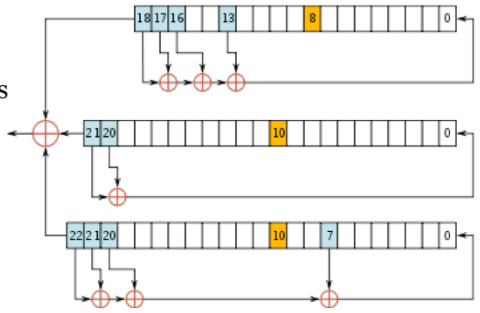
- All sorts of uses
  - Error correcting codes
  - noise generators in CDMA phones
  - good statistical properties for simulations
  - building block in crypto algorithms
- *but not secure in their own right*
  - if you know polynomial, do Gaussian Elimination
  - if you don't, use Berlekamp-Massey, *2n* bits reveals both the state and the feedback polynomial.



The “multiple LFSR” is exactly what you get when you have a larger LFSR. However the particular state and feedback function of the hypothetical larger LFSR is quite complicated when compared to the individual smaller LFSRs.

A5/1

- Algorithm used in GSM phones (where allowed)
- Developed by GSM companies, kept secret
- Uses irregular clocking and multiple registers
- 64 bit key, bit output, stream cipher
- Was never stronger than about 2^{43}
  - and that attack has been extensively optimized, using time-memory tradeoffs (rainbow tables)



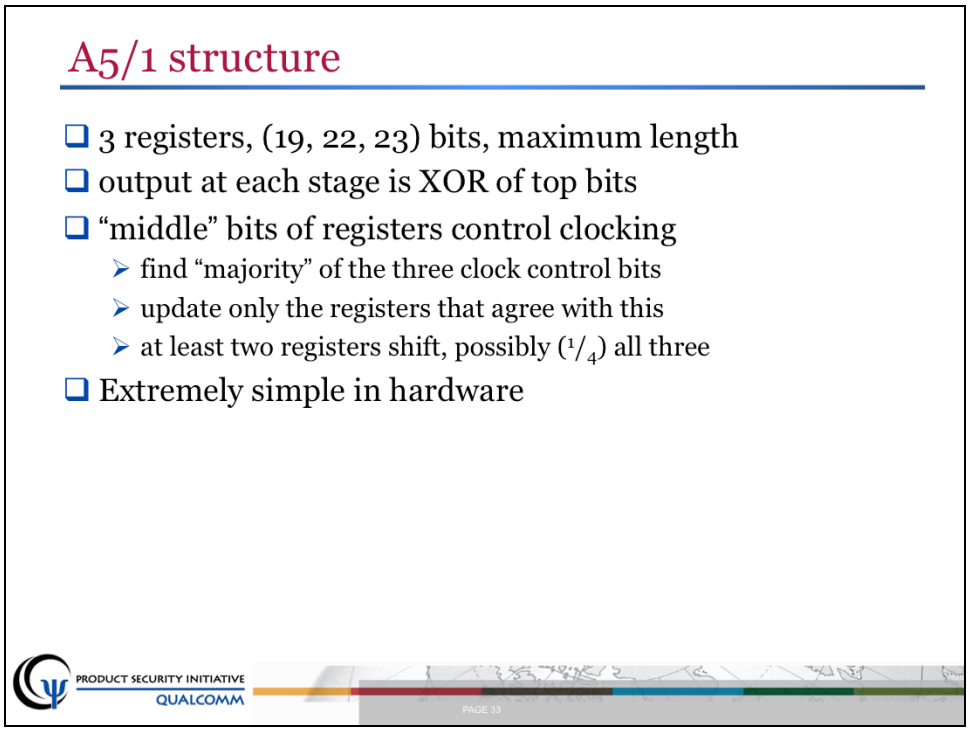
PRODUCT SECURITY INITIATIVE
QUALCOMM

PAGE 32

While the algorithm is capable of accepting a 64 bit key, in actual use in GSM it is only ever fed a *54* bit key (10 of the bits are set to zero). No-one knows a way to exploit this other than brute force, and there are other ways to do break it more efficiently than that.

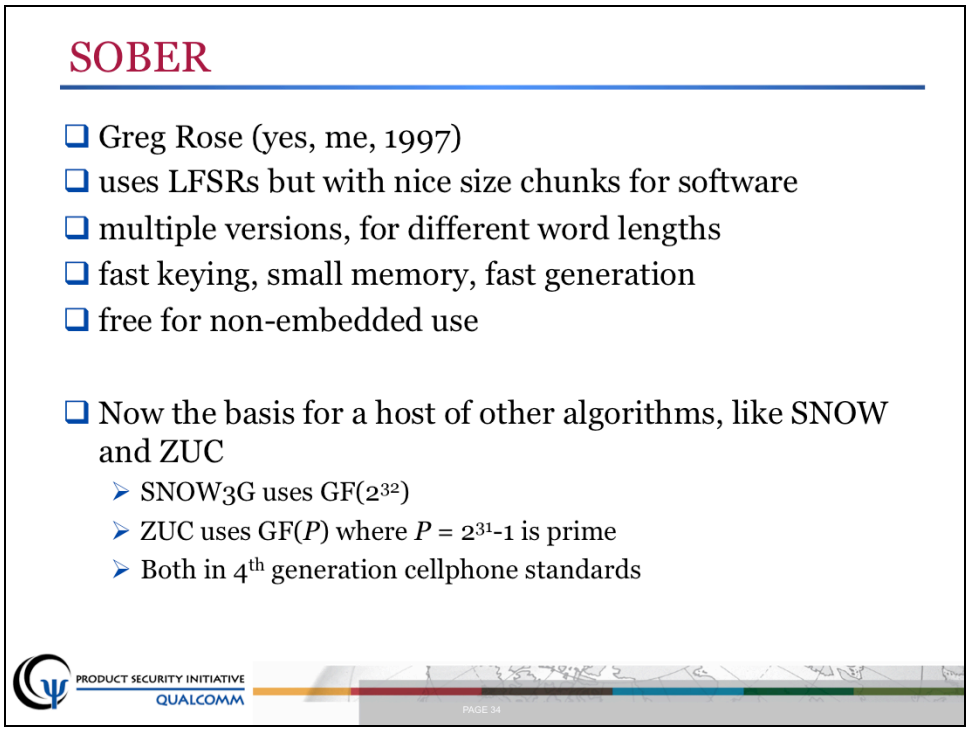
See Biryukov, Shamir and Wagner in Fast Software Encryption, 2000.

Image from Wikipedia.



An early version of A5 appeared in public in the early '90s, but there were a number of things about it which were unknown or incorrect. The correct version (which satisfies the test vectors) was reverse engineered in 1999 by the Smart Card Developers Association, see http://www.scard.org .

There is also an intentionally weakened version called A5/2, in which the clocking is controlled by a fourth shift register and a simple nonlinear filter is added, for which a simple divide-and-conquer attack applies (try every possibility for the fourth register, and derive a set of linear equations relating the unknowns of the original three to the observed outputs. With somewhat more than 64 bits of known plaintext, if you can solve the linear equations, you have with high probability have found the initial state).



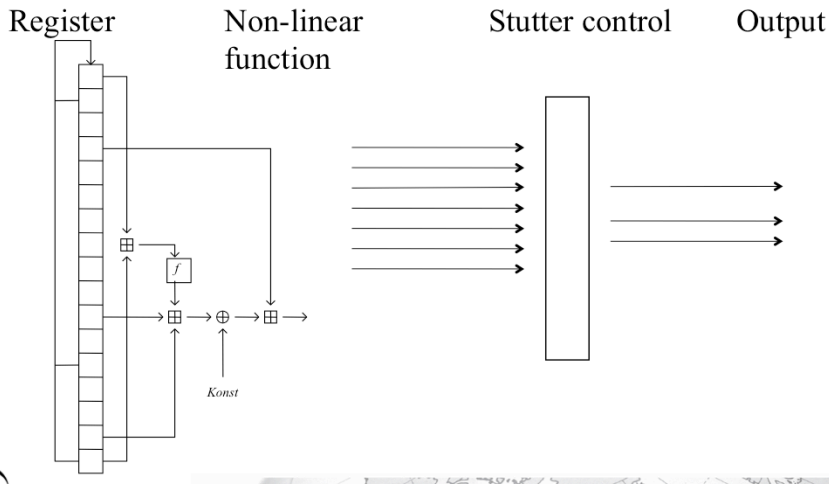
Published in Australian Conference on Information Security and Privacy, 1998. The published one had only linear key mixing, which was a bad mistake, and a new kind of brute force attack was developed (Bleichenbacher, Patel, Sundaramen, FSE' 99). The current version fixes the former, and optimizes against the latter, and the design paper and source code are online at http://www.home.aone.net.au/qualcomm .

SOBER-t16 and –t32 are being considered for standardisation for the European NESSIE project.

LFSRs over GF(2^n)

- Elements of field are word-sized binary polynomials
- Addition operation is XOR, $\oplus$
- Multiplication is poly-mod multiplication, $\otimes$
  - use table lookups
  - only multiply by constants, even better

SOBER block diagram



PRODUCT SECURITY INITIATIVE
QUALCOMM

PAGE 36

Back to the future

- Stream ciphers were well studied up to the mid 1970's
  - DES shifted interest to block ciphers
- Continued deployment through 1990, eg. A5 for cellphones
  - New attacks started to displace them (A5/3 block cipher based)
- But renewed interest in research after AES competition finished
- ECRYPT eSTREAM project started in 2004
- Changed the paradigm for use of stream ciphers

Synchronization

- A5 for cellphones introduced an explicit nonce for each frame
  - but it effectively just modified the key, because the state was small
  - before that stream ciphers either used new keys for each connection or ran continuously
  - FMS attack on WEP exploited related keys (nonce just appended to key) and RC4 bias
- Now, should (securely) support nonce
  - Time-Memory TradeOff (TMTO) implies that the internal state of the stream cipher must be ≥ 1.5 times keylength, nonce must be unpredictable and at least half the keylength

PRODUCT SECURITY INITIATIVE QUALCOMM

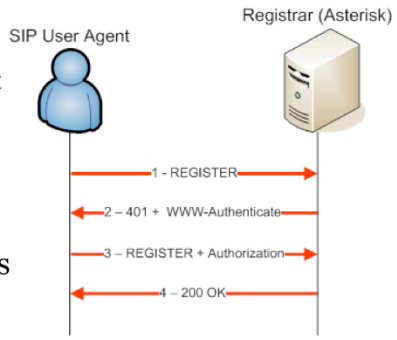
PAGE 38

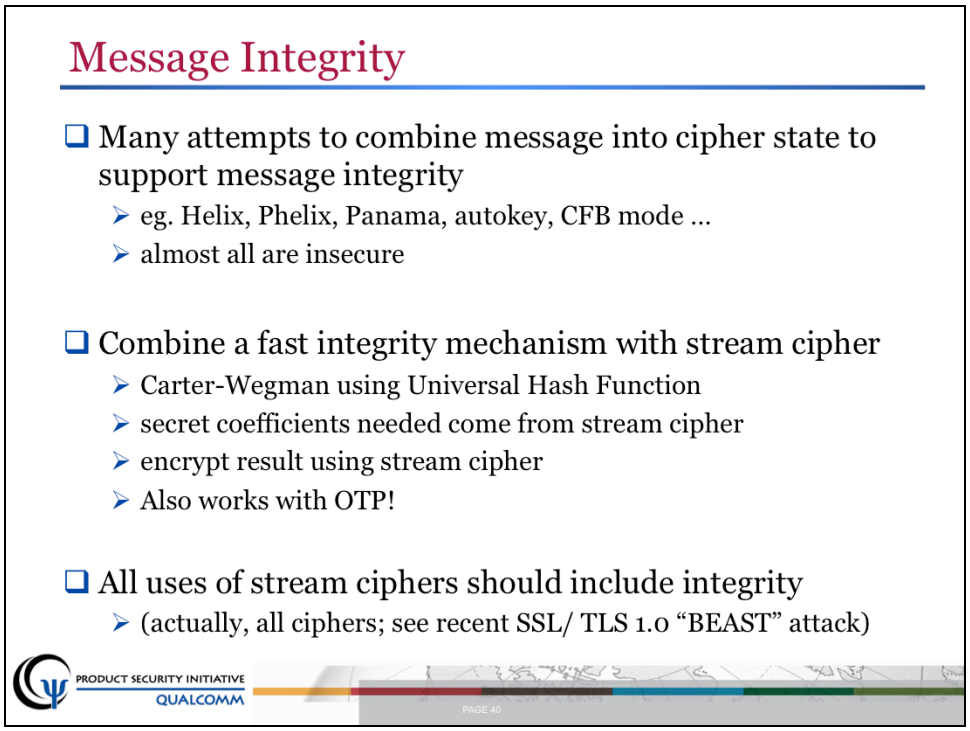
2005/040 Christophe De Cannière, Joseph Lano and Bart Preneel, "Comments on the Rediscovery of Time Memory Data Tradeoffs", pdf, submitted 2005-04-29.

FMS = Fluhrer, Mantin and Shamir “ Weaknesses in the Key Scheduling Algorithm of RC4”

Authentication

- Implicit authentication
  - "Bob (seems to) know the key, must be Bob!"
  - doesn't work (for block ciphers either…)
- Stream ciphers can be used for authentication, but no-one seems to do it
  - Alice->Bob {ID, nonce, keystream}
    Bob->Alice {next keystream}
  - Only works if Bob can detect re-use of nonce! (eg. timestamp based, or counter)
  - or can exchange nonces, more complicated schemes.





On September 23, 2011 researchers Thai Duong and Juliano Rizzo demonstrated a "proof of concept" called BEAST (using a Java Applet to violate "same origin policy" constraints) for a long-known Cipher block chaining (CBC) vulnerability in TLS 1.0. (Wikipedia)

State of the Art

- Best to use combined encryption/integrity primitive
  - GCM, EAX
  - both use AES in Counter Mode – that's a stream cipher!
- Current generation stream ciphers are extremely fast
  - memory for state required can make hardware large
  - fair comparison to block ciphers is hard
- Remaining barrier to use is parallelizability

Done!

- Questions?
- Slides (with notes and hidden slides) available at:
  http://seer-grog.net/streamciphers.pdf