Symmetric ciphers without prior key agreement

Les Hatton

September 10, 2010

A little about this talk

- This is the first talk I have done which apart from its beamer interface used no Microsoft software whatsoever.
- It was constructed on Linux using the Kile interface to Latex and the Beamer package.

Introduction

- Cryptographic model
- Symmetric key cryptography
- Asymmetric key cryptography

Cryptographic Model

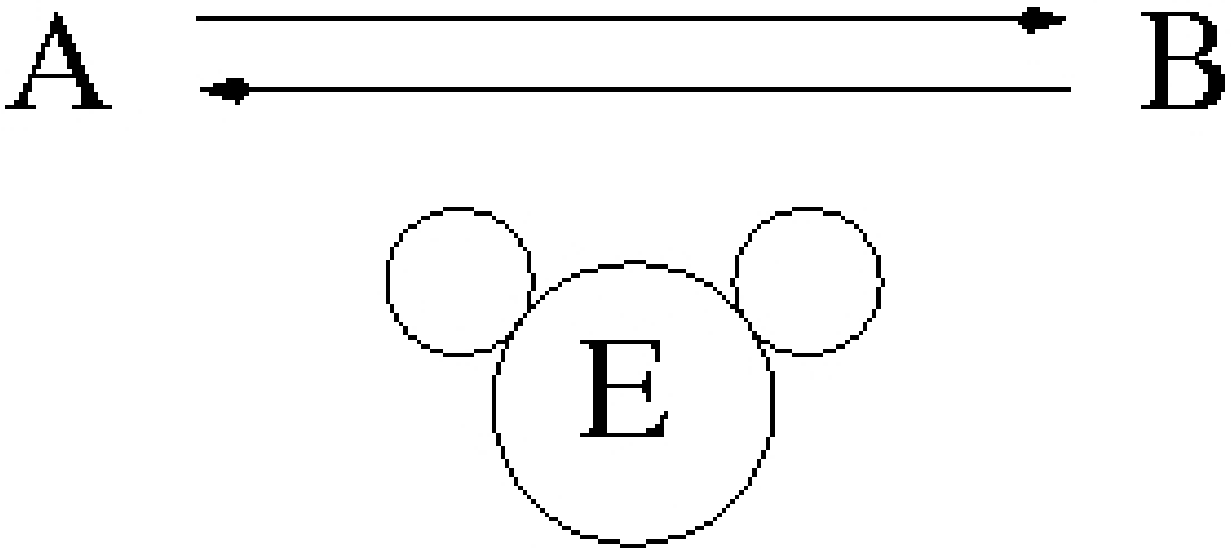

Figure: Alice talks to Bob eavesdropped by Eve

Symmetric key cryptography

- Age-old method.
- Alice secretly tells Bob her key, (one-time pad ...)
- Alice uses this to send encrypted message to Bob.
- Bob decrypts using his secret knowledge

Symmetric schematic

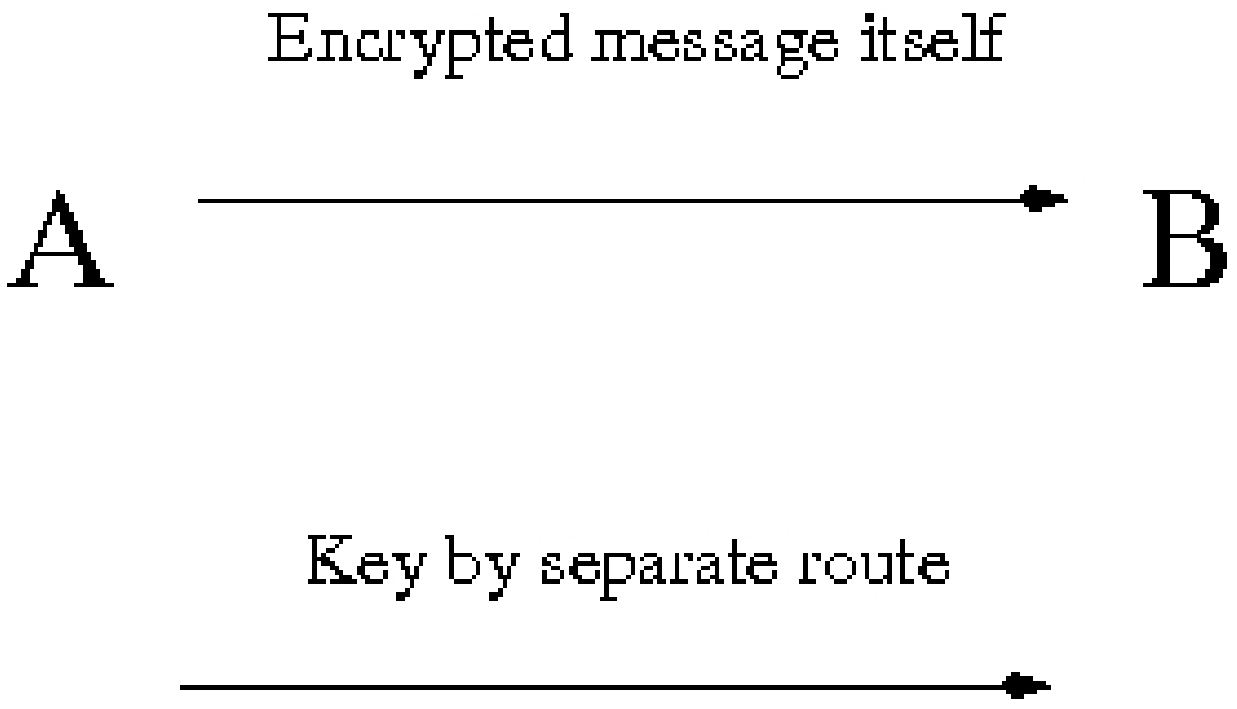

Figure: Alice talks to Bob using symmetric methods

Symmetric key cryptography

- How do you exchange keys without secret knowledge ?
- Solved by Diffie in principle
- Became known as DHM (Diffie-Hellman-Merkle) key exchange.
- Requires commutativity of keys.
- No such keys found and pre-empted by asymmetric methods.

Asymmetric key cryptography

- Invented by Cliff Cocks in early 1970s (Ellis, Cocks and Williamson).
- Rediscovered by Rivest, Shamir and Adleman in 1977 (RSA).
- Depends on intractability of factoring product of large primes and does not require separate sender-receiver key agreement.
- Expensive so often mixed with symmetric algorithms which are much faster.

Asymmetric schematic

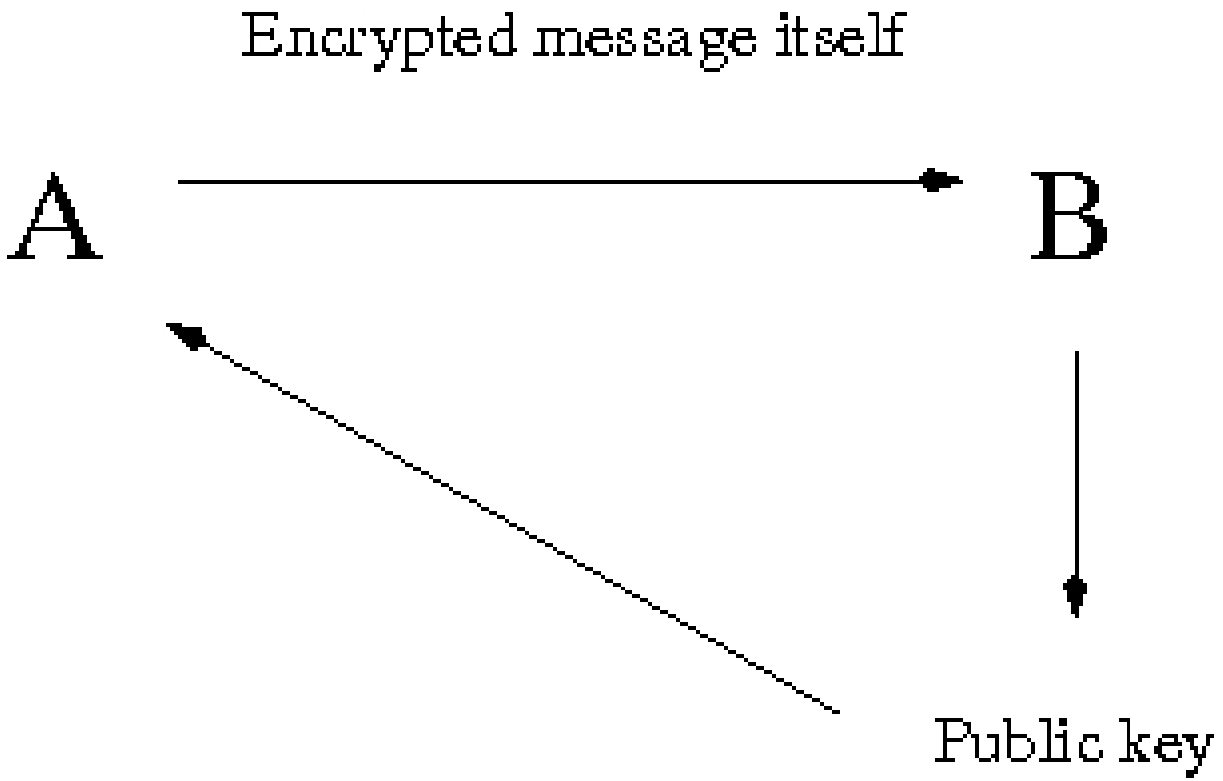

Figure: Alice talks to Bob using asymmetric methods

Motivation

- Asymmetric algorithms can be brute-forced.
- Curiosity to find a symmetric algorithm that doesn't require separate sender-receiver key agreement.

Motivation

- Asymmetric algorithms can be brute-forced.
- Curiosity to find a symmetric algorithm that doesn't require separate sender-receiver key agreement.

Brute-forcing

- The prime factors of a product of two large primes are unique. Once you have factored the product, you have broken the key.
- This has led to larger and larger products.
- If anybody finds a way of doing this efficiently, (and there is no shortage of efforts), then it will severely compromised.

Sender-receiver key agreement

- We are looking for commutative filters with special asymmetric properties.

Mechanics

Alice	Bob
$m \oplus f \rightarrow$	$m \oplus f$
$m \oplus f \oplus g$	$\leftarrow m \oplus f \oplus g$
$m \oplus f \oplus g \oplus f^{-1}$	
$m \oplus g \rightarrow$	$m \oplus g$
	$m \oplus g \oplus g^{-1} = m$

Linear convolution

- We use linear convolution * for $\oplus$

$$e_j = \sum_{k=0}^{L_f-1} f_k m_{j-k} \tag{1}$$

where $e = \{e_0, e_1, ..., e_j, ..., e_{L_f+L_m-1}\}$ is the output.

- However, given m, e how do we find the inverse filter h = f^{-1} such that h*f = i, the identity filter = (1,0,...,0), so that h*f*m recovers the message m ?

Linear convolution

- We use linear convolution * for $\oplus$

$$e_j = \sum_{k=0}^{L_f-1} f_k m_{j-k} \tag{1}$$

where $e = \{e_0, e_1, ..., e_j, ..., e_{L_f+L_m-1}\}$ is the output.

- However, given m, e how do we find the inverse filter h = f^{-1} such that h*f = i, the identity filter = (1,0,...,0), so that h*f*m recovers the message m ?

Linear convolution

- We use linear convolution * for $\oplus$

$$e_j = \sum_{k=0}^{L_f-1} f_k m_{j-k} \tag{1}$$

where $e = \{e_0, e_1, ..., e_j, ..., e_{L_f+L_m-1}\}$ is the output.

- However, given m, e how do we find the inverse filter h = f^{-1} such that h*f = i, the identity filter = (1,0,...,0), so that h*f*m recovers the message m ?

Wiener-Hopf inversion

- First solved by the great American mathematician, Norbert Wiener (1942)
- $$\sum_{t=0}^{L_h} h_t \sum_{k=0}^{L_h+L_f-1} f_{k-t} f_{k-j} = \sum_{k=0}^{L_h+L_f-1} i_k f_{k-j} \qquad (2)$$

  for j = 0, 1, .. , $L_h - 1$. The second sum on the left hand side is the autocorrelation of the input filter whilst the right hand side is the cross-correlation of the input filter with the desired output, i.

Wiener-Hopf inversion

- First solved by the great American mathematician, Norbert Wiener (1942)
- $$\sum_{t=0}^{L_h} h_t \sum_{k=0}^{L_h+L_f-1} f_{k-t} f_{k-j} = \sum_{k=0}^{L_h+L_f-1} i_k f_{k-j} \qquad (2)$$

  for j = 0, 1, .. , $L_h - 1$. The second sum on the left hand side is the autocorrelation of the input filter whilst the right hand side is the cross-correlation of the input filter with the desired output, i.

A filter example

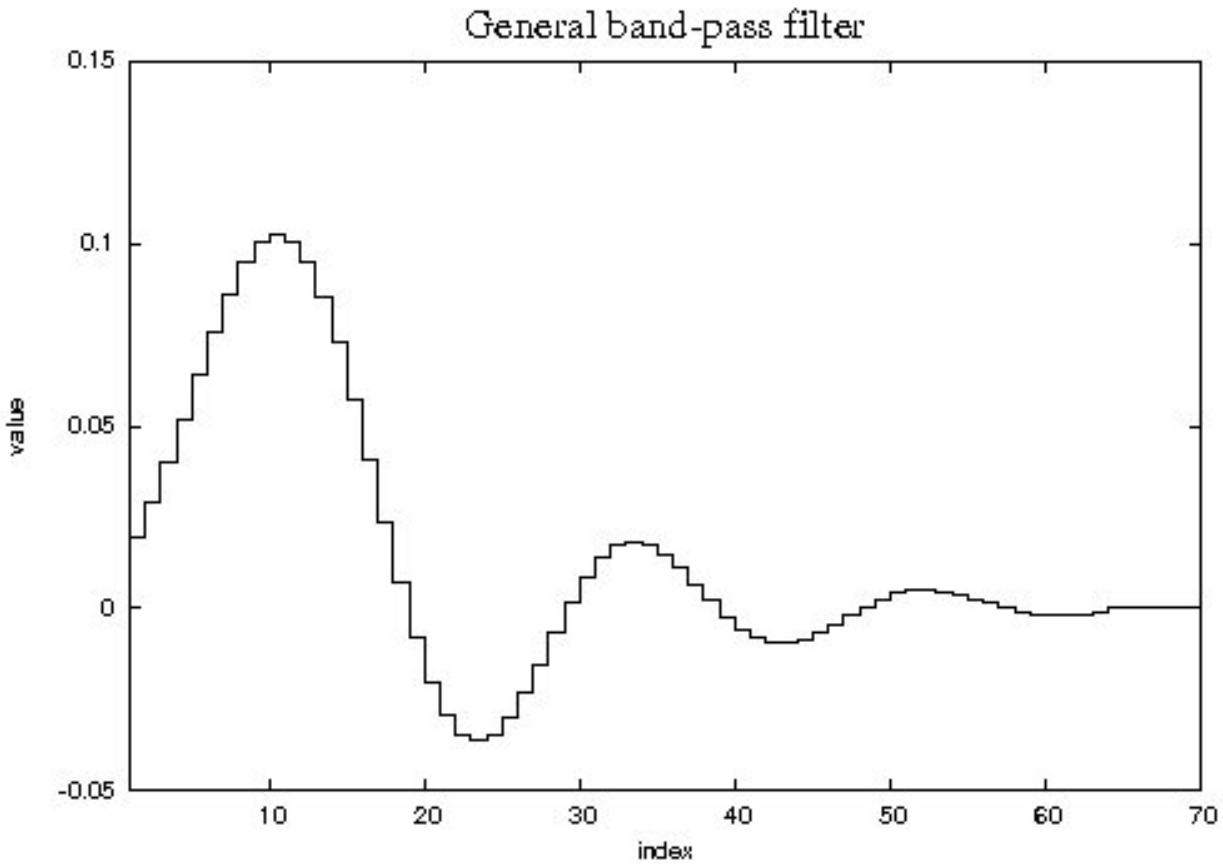

Figure: An example of a convolving filter.

Enter Eve the Eavesdropper

Alice	Bob	Eve
$m * f \longrightarrow$	$m * f$	$m * f$
$m * f * g$	$\longleftarrow m * f * g$	$m * f * g$
$m * f * g * f^{-1}$		
$m * g \longrightarrow$	$m * g$	$m * g$
	$m * g * g^{-1} = m$	

Can Eve unravel m ?

- Eve has three equations and three unknowns !
- But Eve is inverting m*f or m*g whereas Alice and Bob are inverting f and g.
- In the Fourier domain e = m*f is the same as E = MF
- So Alice and Bob are constructing the inverse to F (or G), M = E/F
- Eve is constructing the inverse of MF, so we design the message m to be ill-conditioned, (so M has zeroes in the frequency domain), e.g. Alice constructs m * (1,0,...,-1) * f.

Let's send a message

Message
BANDPASS
0.0000000
18.000000
128.00000
72.000000

Nearly but no cigar

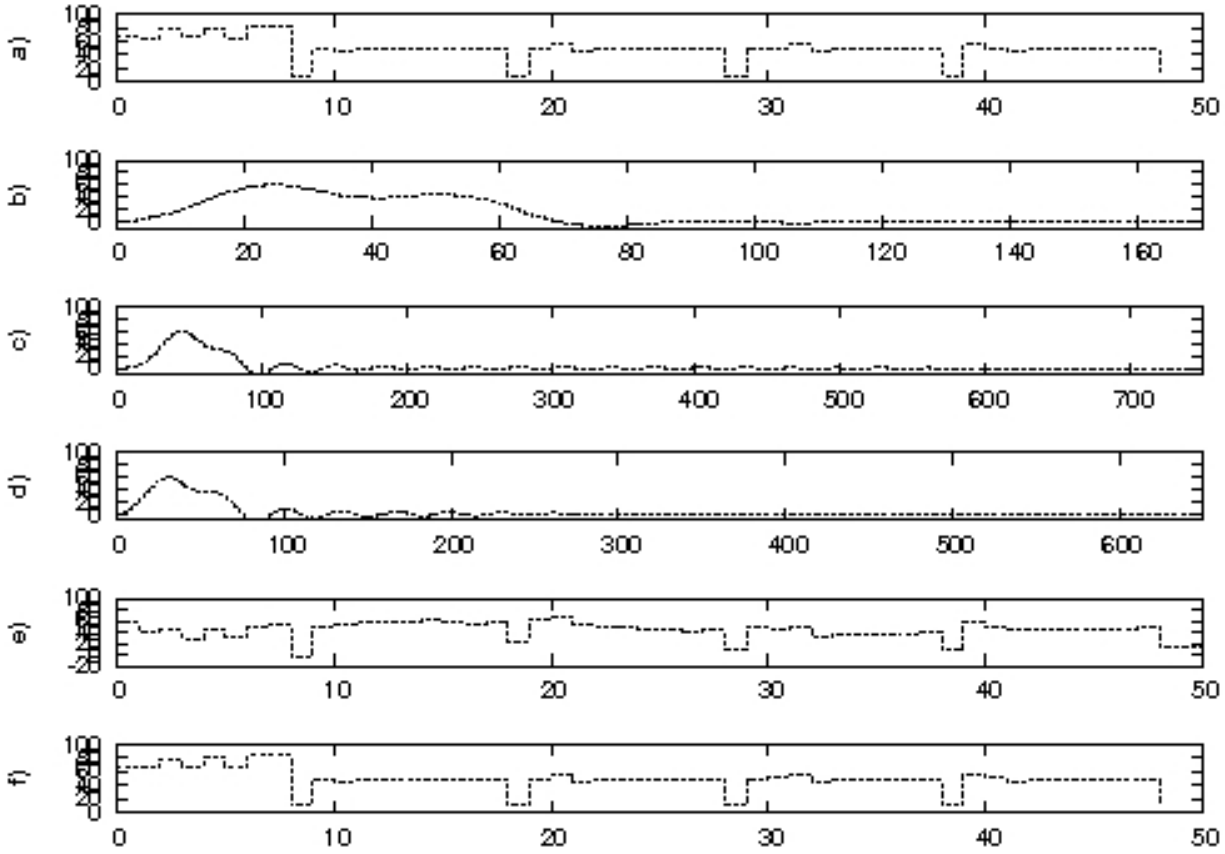

Figure: A complete Wiener-Hopf transaction. a) is the original, b) is m*f, c) is m*f*g, d) is $m * f * g * f^{-1}$, e) is Eve's effort at reconstructing m and f) Bob's effort knowing g.

Just to make sure

- Alice and Bob add in random text starting after the message.
- This means that Eve only sees part of the m*f exchange (up to where r starts) and so on.
- Because the filters are minimum phase, so are their inverses and their correct inversion does not corrupt the original message m. Approximate inversion corrupts m badly.
- This destroys Eve's ability to extract anything useful.

Just to make sure - random text

Alice	Bob
$(m * f) + r \rightarrow$	$(m * f) + r$
$((m * f) + r) * g + s$	$\leftarrow ((m * f) + r) * g + s$
$(((m * f) + r) * g + s) * f^{-1}$	
$m * g + r * g * f^{-1} + s * f^{-1} \rightarrow$	$... * g^{-1}$
	$m + r * f^{-1} + s * f^{-1} * g^{-1}$

Eavesdropping now fails

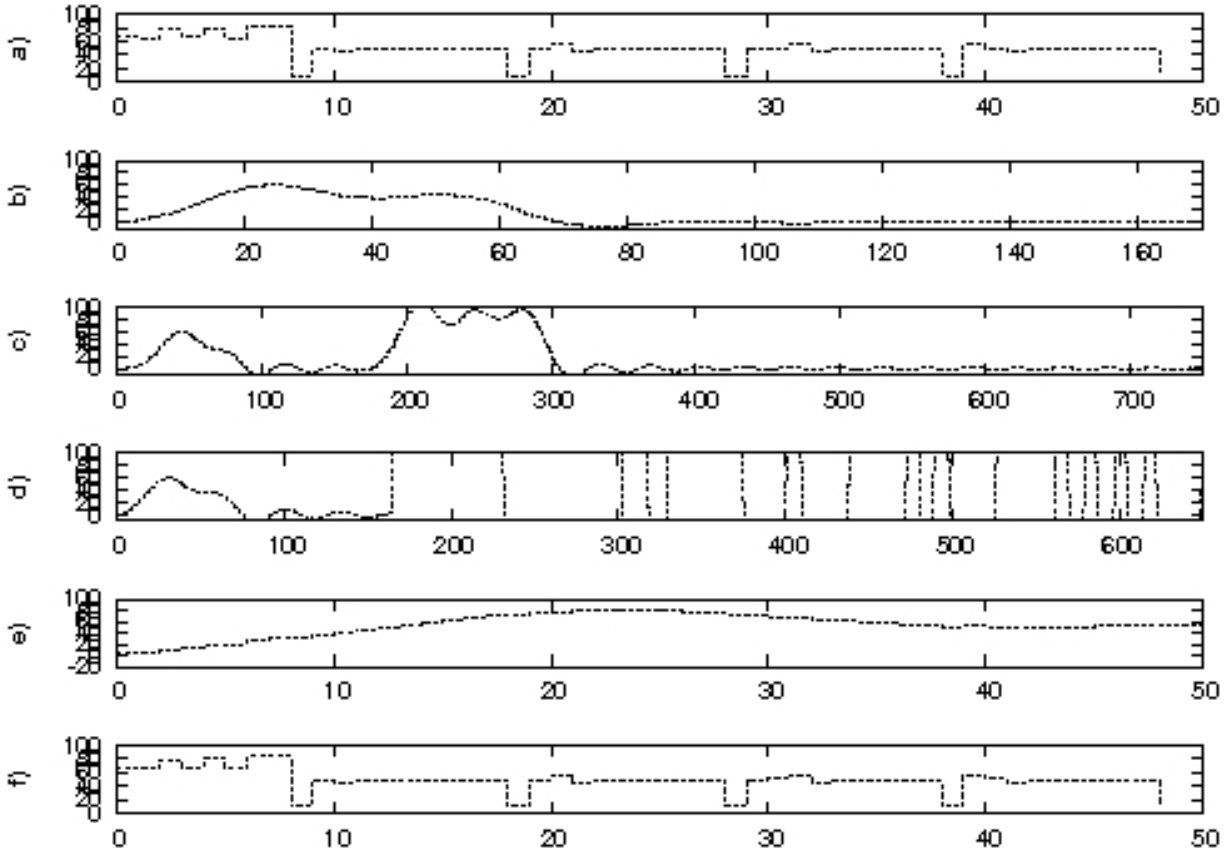

Figure: A complete Wiener-Hopf transaction with confounding randomised text added by both Alice and Bob. a) is the original, b) is m*f+r, c) is (m*f+r)*g+s, d) is $((m * f + r) * g + s) * f^{-1}$, e) is Eve's effort at reconstructing m and f) Bob's effort knowing g.

Attacks

- Brute forcing, non-uniqueness and sensitivity
- Cribs
- Person in the Middle

Brute forcing, non-uniqueness and sensitivity

- Inversion is non-unique !
- For messages short with respect to the filter, redundancy is such that there exist a number of filters, f' such $m * f = m_{false} * f'$.
- For example, there exist filters f, f' such that "999988887777666" * f = "123456785678123 4" * f'.
- Algorithm is very sensitive. Change in 3rd decimal place of just one filter element renders the filtering unreadable.

Brute forcing, non-uniqueness and sensitivity

- Inversion is non-unique !
- For messages short with respect to the filter, redundancy is such that there exist a number of filters, f' such $m * f = m_{false} * f'$.
- For example, there exist filters f, f' such that "999988887777666" * f = "123456785678123 4" * f'.
- Algorithm is very sensitive. Change in 3rd decimal place of just one filter element renders the filtering unreadable.

Brute forcing, non-uniqueness and sensitivity

- Inversion is non-unique !
- For messages short with respect to the filter, redundancy is such that there exist a number of filters, f' such $m * f = m_{false} * f'$.
- For example, there exist filters f, f' such that "9999888877776666" * f = "1234567856781234" * f'.
- Algorithm is very sensitive. Change in 3rd decimal place of just one filter element renders the filtering unreadable.

Brute forcing, non-uniqueness and sensitivity

- Inversion is non-unique !
- For messages short with respect to the filter, redundancy is such that there exist a number of filters, f' such $m * f = m_{false} * f'$.
- For example, there exist filters f, f' such that "999988887777666" * f = "123456785678123 4" * f'.
- Algorithm is very sensitive. Change in 3rd decimal place of just one filter element renders the filtering unreadable.

Cribs

- If Eve ever gets a (m, m*f) pair it is trivial to extract f.
- However, computing f is very fast so we just use a different one for each message.

Cribs

- If Eve ever gets a (m, m*f) pair it is trivial to extract f.
- However, computing f is very fast so we just use a different one for each message.

Person in the middle

- There remains the problem of person in the middle attacks
- These haunt all cryptographic systems (by definition) and modern technology depends on the sanctity of DNS.

Efficiency

Text	Transaction time (s.)
Licence agreement (1.4 Kb.)	0.3
Hamlet (182.6 Kb.)	4.1
King James Bible (4445.3 Kb.)	130.3

Conclusions

- A symmetric algorithm which is robust against eavesdropping has been presented
- Unusually, it is non-unique
- It is probably immune to dictionary, brute force or crib attacks
- Protection from person in the middle attacks still depends on outside agencies.
- More at http://www.leshatton.org/

Conclusions

- A symmetric algorithm which is robust against eavesdropping has been presented
- Unusually, it is non-unique
- It is probably immune to dictionary, brute force or crib attacks
- Protection from person in the middle attacks still depends on outside agencies.
- More at http://www.leshatton.org/

Conclusions

- A symmetric algorithm which is robust against eavesdropping has been presented
- Unusually, it is non-unique
- It is probably immune to dictionary, brute force or crib attacks
- Protection from person in the middle attacks still depends on outside agencies.
- More at http://www.leshatton.org/

Conclusions

- A symmetric algorithm which is robust against eavesdropping has been presented
- Unusually, it is non-unique
- It is probably immune to dictionary, brute force or crib attacks
- Protection from person in the middle attacks still depends on outside agencies.
- More at http://www.leshatton.org/