Public Key Cryptography

Bachelorseminar “Ausgewählte Kapitel der Informatik”

Jan Sprinz

LMU

31.10.2019

Cryptography

cryp · tog · ra · phy

> *"Practice of the enciphering and deciphering of messages in secret code in order to render them unintelligible to all but the intended receiver."*

(Encyclopedia Britannica 2017)

Motivation: Why encrypt anything?

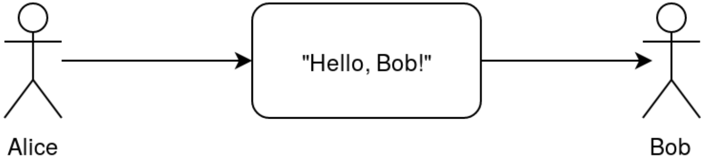

Figure 1: Communication between two parties, “Alice” and “Bob”.

Motivation: Why encrypt anything?

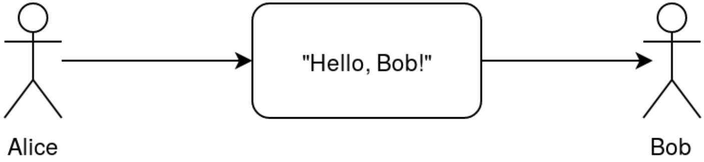

Figure 1: Communication between two parties, “Alice” and “Bob”.

Why Alice and Bob?

- Representing parties “A” and “B” in a transmission
- “Fictional characters commonly used as placeholder names in cryptology” (Wikipedia 2019)
- First introduced by Rivest, Shamir, and Adleman (1978)

Motivation: Why encrypt anything?

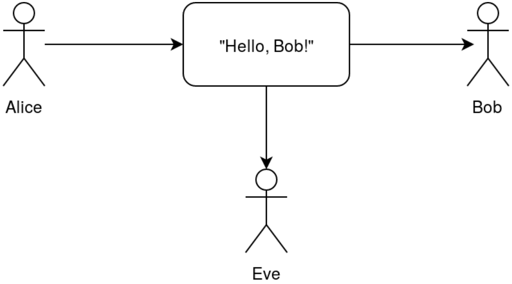

Figure 2: Eavesdropping by a third party, “Eve”, on the communication between two peers, “Alice” and “Bob”. (cf. Wikipedia 2019)

Motivation: Why encrypt anything?

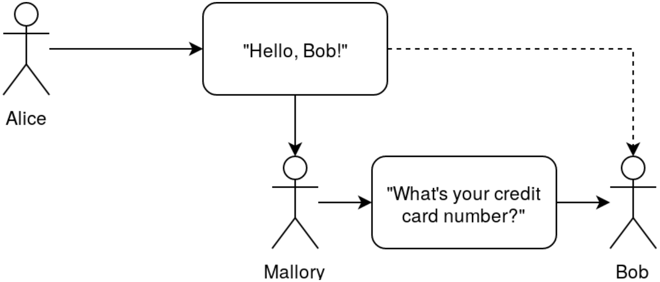

Figure 3: Man-in-the-middle attack: A malicious third party, "Mallory", hijacks the communication between two peers, "Alice" and "Bob". (cf. Wikipedia 2019)

The secure system

Requirements

1. **Confidentiality**: No unauthorized person should be able to read messages.

The secure system

Requirements

1. **Confidentiality**: No unauthorized person should be able to read messages.
2. **Integrity**: No unauthorized party should be able to modify messages.

The secure system

Requirements

1. **Confidentiality**: No unauthorized person should be able to read messages.
2. **Integrity**: No unauthorized party should be able to modify messages.
3. **Authenticity**: All parties need to be verifiable.

The secure system

Requirements

1. **Confidentiality**: No unauthorized person should be able to read messages.
2. **Integrity**: No unauthorized party should be able to modify messages.
3. **Authenticity**: All parties need to be verifiable.
4. **Key Management**: The keys need to be securely created, stored, and distributed.

cf. Ernst, Schmidt, and Beneken (2016), 138

Traditional cipher system

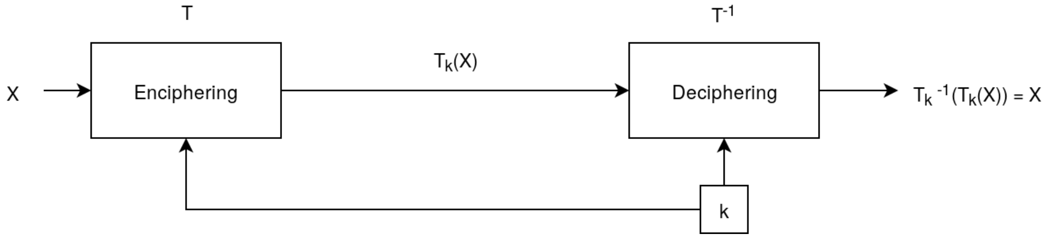

Figure 4: Traditional cipher system for the secure transmission of a message X using a key k and an encryption algorithm T, as well as a decryption algorithm T^{-1}. Own graphic based on Dewdney (2001), 251

Traditional cipher system

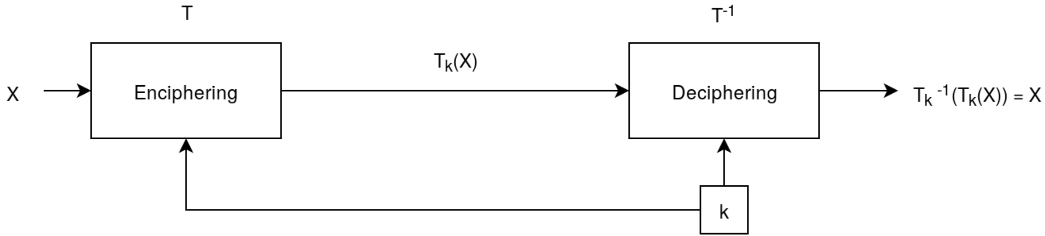

Figure 4: Traditional cipher system for the secure transmission of a message X using a key k and an encryption algorithm T, as well as a decryption algorithm T^{-1}. Own graphic based on Dewdney (2001), 251

Example: caesar code

Replace each letter of the message with the kth letter after it (cf. Ernst, Schmidt, and Beneken 2016, 140).

Traditional cipher system: Example: Caesar code

Example: $X = \textbf{SECRET}$; $k = 4$

Traditional cipher system: Example: Caesar code

Example: $X = \textbf{SECRET}$; $k = 4$

Encryption $T = x_i \rightarrow x_{i+(kMODn)}$

$k=0$	**S**	**E**	**C**	**R**	**E**	**T**
$k=1$	T	F	D	S	F	U
$k=2$	U	G	E	T	G	V
$k=3$	V	H	F	U	H	W
$k=4$	W	I	G	V	I	X

Traditional cipher system: Example: Caesar code

Example: $X =$ **SECRET**; $k = 4$

Encryption $T = x_i \rightarrow x_{i+(kMODn)}$

$k = 0$	**S**	**E**	**C**	**R**	**E**	**T**
$k = 1$	T	F	D	S	F	U
$k = 2$	U	G	E	T	G	V
$k = 3$	V	H	F	U	H	W
$k = 4$	W	I	G	V	I	X

Decryption $T^{-1} = x_i \rightarrow x_{i-(kMODn)}$

$k = 0$	W	I	G	V	I	X
$k = 1$	V	H	F	U	H	W
$k = 2$	U	G	E	T	G	V
$k = 3$	T	F	D	S	F	U
$k = 4$	**S**	**E**	**C**	**R**	**E**	**T**

Limitations of traditional cipher systems

- The key needs to be known to all involved parties **and no one else** $\Rightarrow$ the key needs to be communicated over a secure channel

Limitations of traditional cipher systems

- The key needs to be known to all involved parties **and no one else** $\Rightarrow$ the key needs to be communicated over a secure channel
- The system does not scale

Limitations of traditional cipher systems

- The key needs to be known to all involved parties **and no one else** $\Rightarrow$ the key needs to be communicated over a secure channel
- The system does not scale
- The key is a single point of failure, and is stored in multiple locations

Public Key Cryptography: Concept

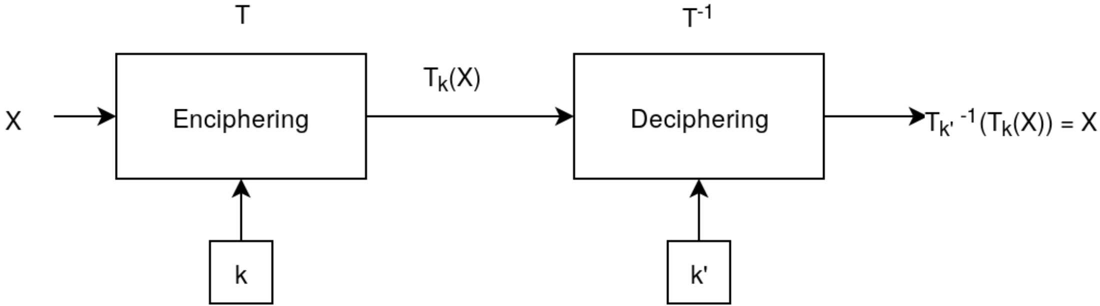

Figure 5: Public key cipher system. Own graphic based on Diffie and Hellman (1976), 647

Usecase: Signing

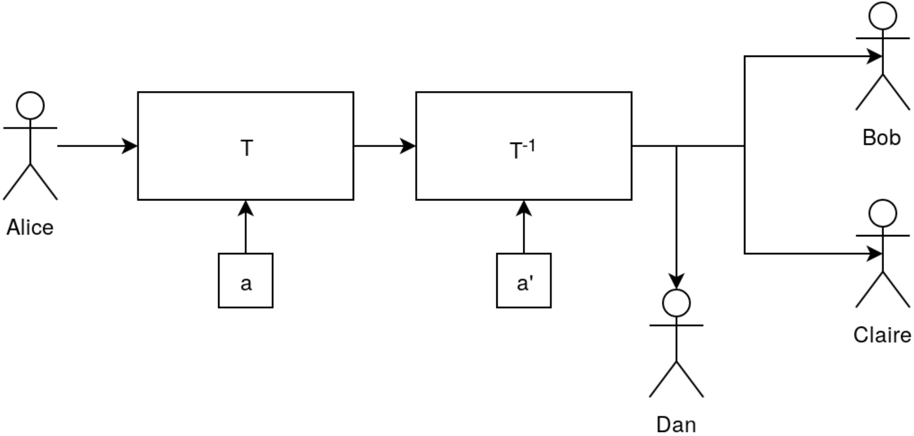

Figure 6: "Alice" encrypts a message with her private key a. Everyone receiving the message can verify its authenticity by decrypting it with her public key a'.

Usecase: Secure communication

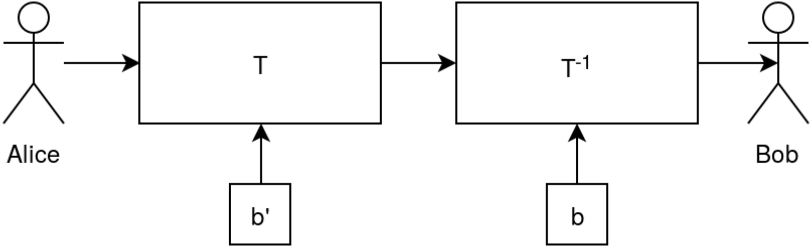

Figure 7: “Alice” encrypts a message with Bob’s public key b'. Only Bob can decrypt it with his private key b.

Usecase: Signed secure communication

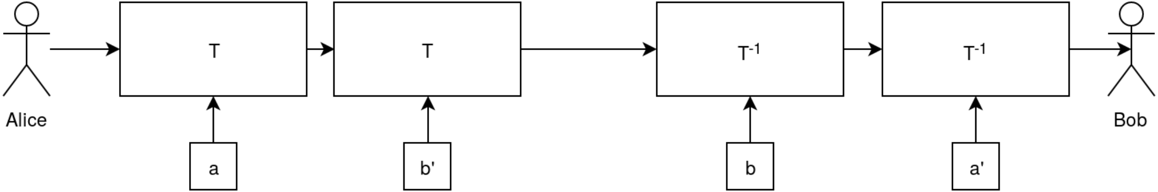

Figure 8: "Alice" encrypts a message with her private key a and Bob's public key b'. Bob can verify the authenticity of the message by decrypting with Alice's public key and a' and his private key b.

Requirements and challenges

Computing private key k and public key k'

- k and k' need to be easy to generate

Requirements and challenges

Computing private key k and public key k'

- k and k' need to be easy to generate
- k' must be easy to compute from k

Requirements and challenges

Computing private key k and public key k'

- k and k' need to be easy to generate
- k' must be easy to compute from k
- k must be difficult to compute from k'

cf. Dewdney (2001), 252

Requirements and challenges

Computing private key k and public key k'

- k and k' need to be easy to generate
- k' must be easy to compute from k
- k must be difficult to compute from k'

cf. Dewdney (2001), 252

Avoiding security by obscurity

"The reader is urged to find a way to 'break' the system. Once the method has withstood all attacks for a sufficient length of time it may be used with a reasonable amount of confidence."
(Rivest, Shamir, and Adleman 1978, 126)

Requirements and challenges

Computing private key k and public key k'

- k and k' need to be easy to generate
- k' must be easy to compute from k
- k must be difficult to compute from k'

cf. Dewdney (2001), 252

Encryption is broken if...

- The private key is leaked

Avoiding security by obscurity

"The reader is urged to find a way to 'break' the system. Once the method has withstood all attacks for a sufficient length of time it may be used with a reasonable amount of confidence."
(Rivest, Shamir, and Adleman 1978, 126)

Requirements and challenges

Computing private key k and public key k'

- k and k' need to be easy to generate
- k' must be easy to compute from k
- k must be difficult to compute from k'

cf. Dewdney (2001), 252

Avoiding security by obscurity

"The reader is urged to find a way to 'break' the system. Once the method has withstood all attacks for a sufficient length of time it may be used with a reasonable amount of confidence."
(Rivest, Shamir, and Adleman 1978, 126)

Encryption is broken if...

- The private key is leaked
- The encryption system itself is cracked

cf. Dewdney (2001), 255

Requirements and challenges

Computing private key k and public key k'

- k and k' need to be easy to generate
- k' must be easy to compute from k
- k must be difficult to compute from k'

cf. Dewdney (2001), 252

Avoiding security by obscurity

"The reader is urged to find a way to 'break' the system. Once the method has withstood all attacks for a sufficient length of time it may be used with a reasonable amount of confidence."
(Rivest, Shamir, and Adleman 1978, 126)

Encryption is broken if...

- The private key is leaked
- The encryption system itself is cracked

cf. Dewdney (2001), 255

Our cryptosystem is broken if...

- Our problem is not *NP*-complete

Requirements and challenges

Computing private key k and public key k'

- k and k' need to be easy to generate
- k' must be easy to compute from k
- k must be difficult to compute from k'

cf. Dewdney (2001), 252

Avoiding security by obscurity

"The reader is urged to find a way to 'break' the system. Once the method has withstood all attacks for a sufficient length of time it may be used with a reasonable amount of confidence."
(Rivest, Shamir, and Adleman 1978, 126)

Encryption is broken if...

- The private key is leaked
- The encryption system itself is cracked

cf. Dewdney (2001), 255

Our cryptosystem is broken if...

- Our problem is not NP-complete
- Someone proves that $P == NP$

cf. Dewdney (2001), 255

RSA

cf. Dewdney (2001), 255

Underlying principle

- based on the factorization problem: find a non-trivial factor for an n-bit number

RSA

cf. Dewdney (2001), 255

Underlying principle

- based on the factorization problem: find a non-trivial factor for an n-bit number

In practice

- the keys are generated from two prime factors p and q

RSA

cf. Dewdney (2001), 255

Underlying principle

- based on the factorization problem: find a non-trivial factor for an n-bit number

In practice

- the keys are generated from two prime factors p and q
- the product $n = pq$ becomes the first part of the public key

RSA

cf. Dewdney (2001), 255

Underlying principle

- based on the factorization problem: find a non-trivial factor for an n-bit number

In practice

- the keys are generated from two prime factors p and q
- the product $n = pq$ becomes the first part of the public key
- second part of the public key: $e \begin{cases} 1 < e < \phi(n) \\ \text{coprime of } n \text{ and } \phi(n) \end{cases}$ with $\phi(n) = (p-1)(q-1)$

RSA

cf. Dewdney (2001), 255

Underlying principle

- based on the factorization problem: find a non-trivial factor for an n-bit number

In practice

- the keys are generated from two prime factors p and q
- the product $n = pq$ becomes the first part of the public key
- second part of the public key: $e \begin{cases} 1 < e < \phi(n) \\ \text{coprime of } n \text{ and } \phi(n) \end{cases}$ with $\phi(n) = (p-1)(q-1)$
- *coprimes*: set of integers that only share 1 as a factor

RSA

cf. Dewdney (2001), 255

Underlying principle

- based on the factorization problem: find a non-trivial factor for an n-bit number

In practice

- the keys are generated from two prime factors p and q
- the product $n = pq$ becomes the first part of the public key
- second part of the public key: $e \begin{cases} 1 < e < \phi(n) \\ \text{coprime of } n \text{ and } \phi(n) \end{cases}$ with $\phi(n) = (p-1)(q-1)$
- *coprimes*: set of integers that only share 1 as a factor
- a message $m < n$ is encrypted using the following formula $c = m^e \text{ MOD } n$

RSA

cf. Dewdney (2001), 255

Underlying principle

- based on the factorization problem: find a non-trivial factor for an n-bit number

In practice

- the keys are generated from two prime factors p and q
- the product $n = pq$ becomes the first part of the public key
- second part of the public key: $e \begin{cases} 1 < e < \phi(n) \\ \text{coprime of } n \text{ and } \phi(n) \end{cases}$ with $\phi(n) = (p-1)(q-1)$
- *coprimes*: set of integers that only share 1 as a factor
- a message $m < n$ is encrypted using the following formula $c = m^e \text{ MOD } n$
- the private key is the integer $d : 1 = ed \text{ MOD } \phi(n)$

RSA

cf. Dewdney (2001), 255

Underlying principle

- based on the factorization problem: find a non-trivial factor for an n-bit number

In practice

- the keys are generated from two prime factors p and q
- the product $n = pq$ becomes the first part of the public key
- second part of the public key: $e \begin{cases} 1 < e < \phi(n) \\ \text{coprime of } n \text{ and } \phi(n) \end{cases}$ with $\phi(n) = (p-1)(q-1)$
- *coprimes*: set of integers that only share 1 as a factor
- a message $m < n$ is encrypted using the following formula $c = m^e \text{ MOD } n$
- the private key is the integer $d : 1 = ed \text{ MOD } \phi(n)$
- the message can be decrypted by computing $c^d \text{ MOD } n = m$.

RSA: Example: Generate key pair

1. Two prime numbers $p = 2$, $q = 7$

RSA: Example: Generate key pair

1. Two prime numbers $p = 2$, $q = 7$
2. Calculate $n = pq = 2 * 7 = 14$

RSA: Example: Generate key pair

1. Two prime numbers $p = 2$, $q = 7$
2. Calculate $n = pq = 2 * 7 = 14$
3. Calculate $\phi(n)$, the number of coprimes of n: $1, 3, 5, 9, 11, 13$

RSA: Example: Generate key pair

1. Two prime numbers $p = 2$, $q = 7$
2. Calculate $n = pq = 2 * 7 = 14$
3. Calculate $\phi(n)$, the number of coprimes of n: $1, 3, 5, 9, 11, 13$

- $\phi(n) = \phi(14) = (p - 1)(q - 1) = (2 - 1)(7 - 1) = 6$

RSA: Example: Generate key pair

1. Two prime numbers $p = 2,\ q = 7$
2. Calculate $n = pq = 2 * 7 = 14$
3. Calculate $\phi(n)$, the number of coprimes of n: $1, 3, 5, 9, 11, 13$
   - $\phi(n) = \phi(14) = (p-1)(q-1) = (2-1)(7-1) = 6$
4. Calculate $e \begin{cases} 1 < e < \phi(n) \\ \text{coprime of } n \text{ and } \phi(n) \end{cases} \Rightarrow e = 5$

RSA: Example: Generate key pair

1. Two prime numbers $p = 2$, $q = 7$
2. Calculate $n = pq = 2 * 7 = 14$
3. Calculate $\phi(n)$, the number of coprimes of n: $1, 3, 5, 9, 11, 13$
   - $\phi(n) = \phi(14) = (p-1)(q-1) = (2-1)(7-1) = 6$
4. Calculate $e \begin{cases} 1 < e < \phi(n) \\ \text{coprime of } n \text{ and } \phi(n) \end{cases} \Rightarrow e = 5$
5. Choose $d : 1 = ed \text{ MOD } \phi(n)$, for example 11

RSA: Example: Generate key pair

1. Two prime numbers $p = 2,\ q = 7$
2. Calculate $n = pq = 2 * 7 = 14$
3. Calculate $\phi(n)$, the number of coprimes of n: $1, 3, 5, 9, 11, 13$
   - $\phi(n) = \phi(14) = (p-1)(q-1) = (2-1)(7-1) = 6$
4. Calculate $e \begin{cases} 1 < e < \phi(n) \\ \text{coprime of } n \text{ and } \phi(n) \end{cases} \Rightarrow e = 5$
5. Choose $d : 1 = ed \text{ MOD } \phi(n)$, for example 11

p	***q***	***d***	***e***	***n***
2	7	11	5	14

RSA: Example: Encrypt and Decrypt

p	q	d	$\boldsymbol{e}$	$\boldsymbol{n}$	m	$\boldsymbol{c}$
2	7	11	5	14	$C = 3$	$E = 5$

Encrypt

$$c = m^e \text{ MOD } n$$

RSA: Example: Encrypt and Decrypt

p	q	d	$\boldsymbol{e}$	$\boldsymbol{n}$	m	$\boldsymbol{c}$
2	7	11	5	14	$C = 3$	$E = 5$

Encrypt

$$c = m^e \text{ MOD } n$$

$$c = 3^5 \text{ MOD } 14 = 5 = E$$

RSA: Example: Encrypt and Decrypt

p	q	d	$\boldsymbol{e}$	$\boldsymbol{n}$	m	$\boldsymbol{c}$
2	7	11	5	14	$C = 3$	$E = 5$

Encrypt

$$c = m^e \text{ MOD } n$$

$$c = 3^5 \text{ MOD } 14 = 5 = E$$

Decrypt

$$m = c^d \text{ MOD } n$$

RSA: Example: Encrypt and Decrypt

p	q	d	$\boldsymbol{e}$	$\boldsymbol{n}$	m	$\boldsymbol{c}$
2	7	11	5	14	$C = 3$	$E = 5$

Encrypt

$$c = m^e \text{ MOD } n$$

$$c = 3^5 \text{ MOD } 14 = 5 = E$$

Decrypt

$$m = c^d \text{ MOD } n$$

$$m = 5^{11} \text{ MOD } 14 = 3 = C$$

RSA: Is it secure?

RSA: Is it secure?

No

- NP-completeness has never been proven, so there might highly efficient algorithms to solve the factorization problem

RSA: Is it secure?

No

- NP-completeness has never been proven, so there might highly efficient algorithms to solve the factorization problem
- Quantum computers allow for much more efficient factorization

RSA: Is it secure?

No

- NP-completeness has never been proven, so there might highly efficient algorithms to solve the factorization problem
- Quantum computers allow for much more efficient factorization
- Computers are getting faster exponentially (*moore's law*), so brute-forcing the key becomes easier

RSA: Is it secure?

No

- NP-completeness has never been proven, so there might highly efficient algorithms to solve the factorization problem
- Quantum computers allow for much more efficient factorization
- Computers are getting faster exponentially (*moore's law*), so brute-forcing the key becomes easier

Yes

- There's an infinite number of primes, so bigger factors can be used

RSA: Is it secure?

No

- NP-completeness has never been proven, so there might highly efficient algorithms to solve the factorization problem
- Quantum computers allow for much more efficient factorization
- Computers are getting faster exponentially (*moore's law*), so brute-forcing the key becomes easier

Yes

- There's an infinite number of primes, so bigger factors can be used
- Algorithms are still not efficient enough to make cracking encryption profitable

RSA: Is it secure?

No

- NP-completeness has never been proven, so there might highly efficient algorithms to solve the factorization problem
- Quantum computers allow for much more efficient factorization
- Computers are getting faster exponentially (*moore's law*), so brute-forcing the key becomes easier

Yes

- There's an infinite number of primes, so bigger factors can be used
- Algorithms are still not efficient enough to make cracking encryption profitable
- Quantum computers are still very experimental

RSA: Is it secure?

No

- NP-completeness has never been proven, so there might highly efficient algorithms to solve the factorization problem
- Quantum computers allow for much more efficient factorization
- Computers are getting faster exponentially (*moore's law*), so brute-forcing the key becomes easier

Yes

- There's an infinite number of primes, so bigger factors can be used
- Algorithms are still not efficient enough to make cracking encryption profitable
- Quantum computers are still very experimental
- In practice, bugs in implementations are a more likely attack vector

cf. Ernst, Schmidt, and Beneken (2016), 164

Bibliography

Dewdney, Alexander K. 2001. *The (New) Turing Omnibus: 66 Excurions in Computer Science*. 1. paperbacks ed. Holt Paperback. New York, NY: Freemann.

Diffie, W., and M. Hellman. 1976. "New Directions in Cryptography." *IEEE Transactions on Information Theory* 22 (6): 644–54.

Encyclopedia Britannica. 2017. "Cryptography." April 13, 2017. https://www.britannica.com/topic/cryptography.

Ernst, Hartmut, Jochen Schmidt, and Gerd Hinrich Beneken. 2016. *Grundkurs Informatik*. 6. Auflage. Lehrbuch. Wiesbaden: Springer Vieweg.

Rivest, R. L., A. Shamir, and L. Adleman. 1978. "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems." *Commun. ACM* 21 (2): 120–26.

Wikipedia. 2019. "Alice and Bob." *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Alice_and_Bob&oldid=922042581.