Public-key and symmetric-key cryptology

Tanja Lange

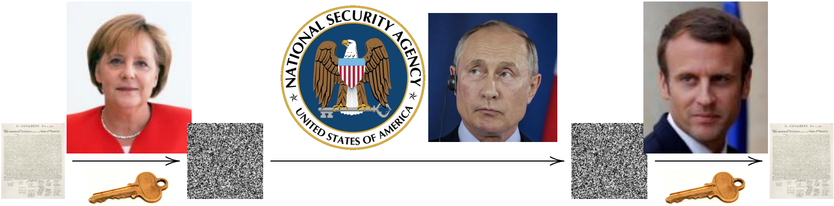
Eindhoven University of Technology

2WF80: Introduction to Cryptology

If you have a secret channel . . .

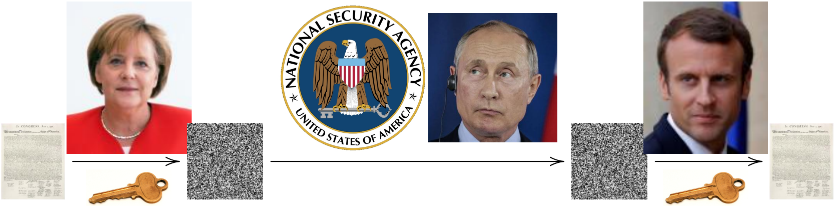
. . . you can agree on a shared key . . .

. . . and use that key to encrypt and authenticate



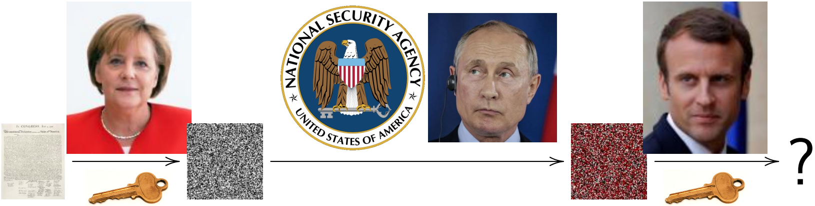
- Symmetric-key cryptography:
  Alice and Bob share a secret key 🔑.
- Prerequisite: Eve doesn't know 🔑.
- Alice and Bob exchange any number of messages.
- Encryption takes plaintext m and produces ciphertext c, decryption takes c and produces m so that $\mathrm{Dec}(\mathrm{Enc}(m)) = m$.
- Security goal #1: **Confidentiality** despite Eve's espionage.

. . . and use that key to encrypt and authenticate



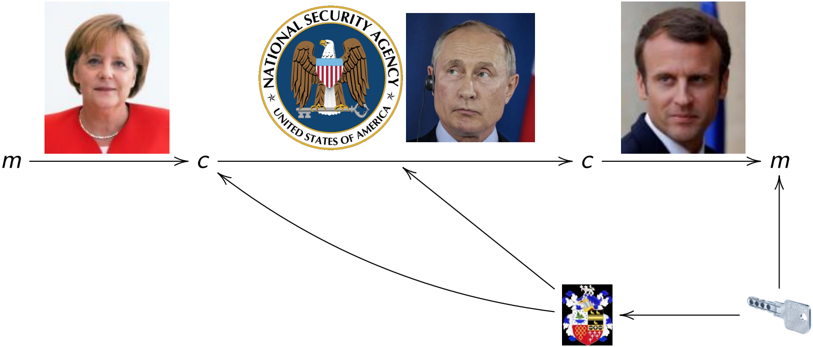
- Symmetric-key cryptography:
  Alice and Bob share a secret key 🔑.
- Prerequisite: Eve doesn't know 🔑.
- Alice and Bob exchange any number of messages.
- Encryption takes plaintext m and produces ciphertext c,
  decryption takes c and produces m so that $\text{Dec}(\text{Enc}(m)) = m$.
- Security goal #1: **Confidentiality** despite Eve's espionage.
- Security goal #2: **Integrity**, i.e., recognizing Eve's sabotage.
- Security goal #3: **Authenticity**, i.e., recognizing Eve impersonating.

. . . and use that key to encrypt and authenticate



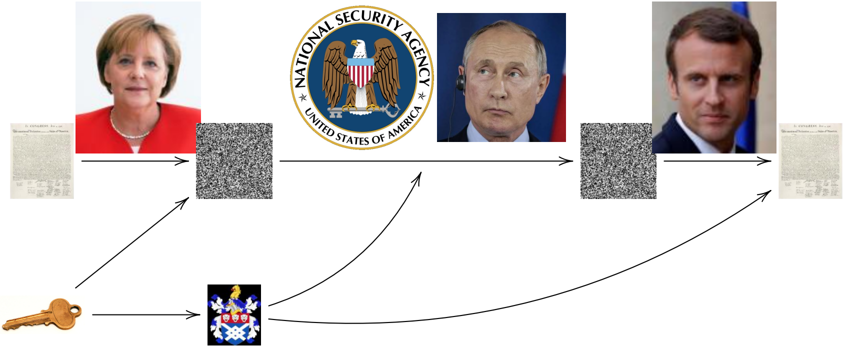
- Symmetric-key cryptography:
  Alice and Bob share a secret key 🔑.
- Prerequisite: Eve doesn't know 🔑.
- Alice and Bob exchange any number of messages.
- Encryption takes plaintext m and produces ciphertext c, decryption takes c and produces m so that $\mathrm{Dec}(\mathrm{Enc}(m)) = m$.
- Security goal #1: **Confidentiality** despite Eve's espionage.
- Security goal #2: **Integrity**, i.e., recognizing Eve's sabotage.
- Security goal #3: **Authenticity**, i.e., recognizing Eve impersonating.
- Decryption fails for invalid ciphertexts.
  (This needs a definition of what "invalid" means).

Public-key encryption



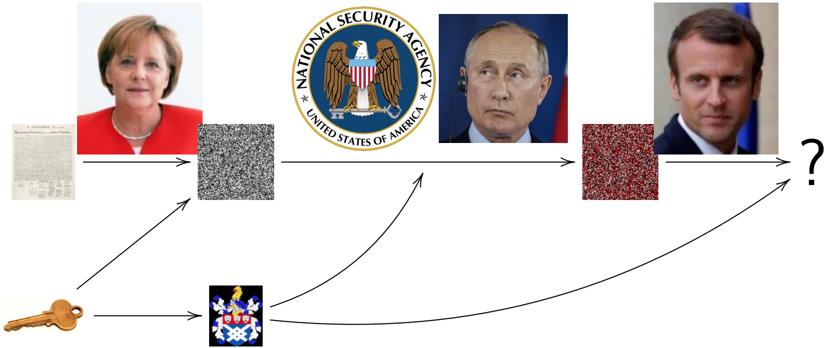
- Public-key cryptography: each user has two keys, a public key and a private key.
- Everybody, including Eve, knows the public key.
- Secure systems make it computationally impossible to recover the private key from the public key.
- Alice uses Bob's public key $K =$ to encrypt plaintext m.
- Bob uses his private key $k =$ to decrypt ciphertext c.

Public-key signatures



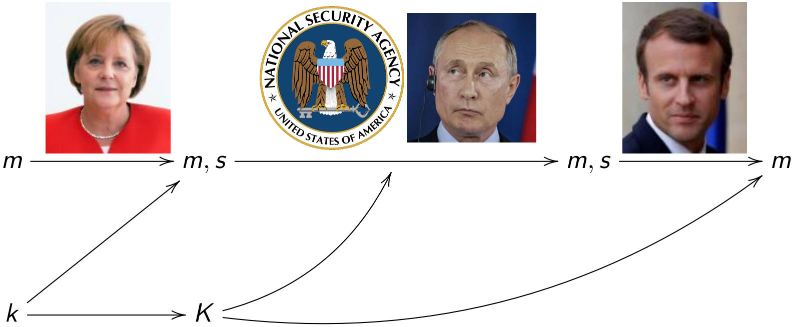
- Prerequisite: Alice has a private key 🔑 and public key 🛡.
- Prerequisite: Everyone knows 🛡 as belonging to Alice.
- Alice signs messages using 🔑. Other people verify using 🛡.

Public-key signatures



- Prerequisite: Alice has a private key 🔑 and public key 🛡.
- Prerequisite: Everyone knows 🛡 as belonging to Alice.
- Alice signs messages using 🔑. Other people verify using 🛡.
- Security goals: Integrity and authenticity.
- Nobody can produce signatures valid under 🛡 without 🔑.
- Modifications to signed message get caught.

Public-key signatures

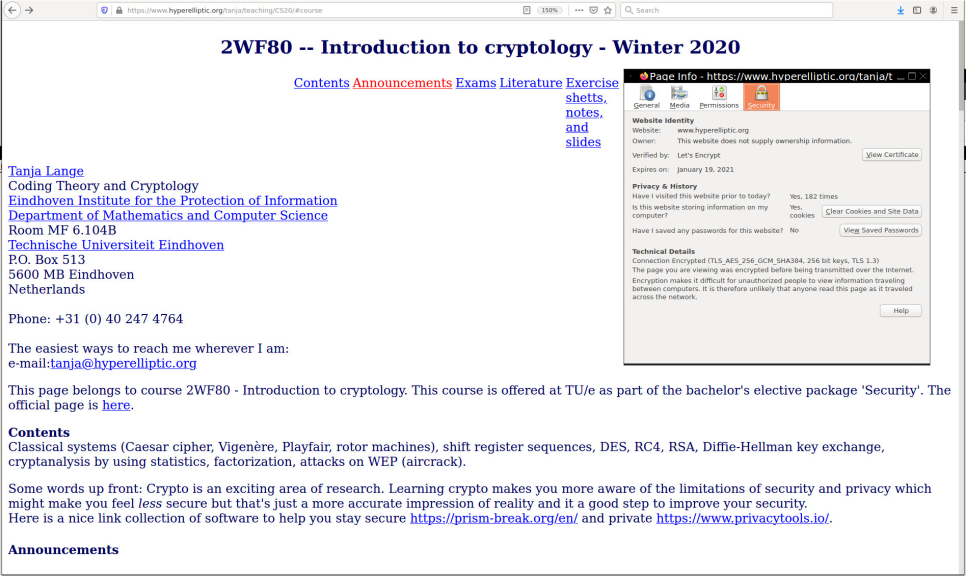


$$\begin{tikzcd}
m \arrow[r] & m, s \arrow[rr] & & m, s \arrow[r] & m \\
k \arrow[r] \arrow[ur] & K \arrow[ur, bend right] \arrow[urrr, bend right] & & &
\end{tikzcd}$$

Private key $k =$, public key $K =$.

Older systems, and that includes PGP/GPG, send m, s, i.e., let the user see m before/without verifying S.

Modern systems send a signed message s and the verification algorithm returns m or "invalid".



2WF80 -- Introduction to cryptology - Winter 2020

Contents Announcements Exams Literature Exercise shetts, notes, and slides

Tanja Lange
Coding Theory and Cryptology
Eindhoven Institute for the Protection of Information
Department of Mathematics and Computer Science
Room MF 6.104B
Technische Universiteit Eindhoven
P.O. Box 513
5600 MB Eindhoven
Netherlands

Phone: +31 (0) 40 247 4764

The easiest ways to reach me wherever I am:
e-mail:tanja@hyperelliptic.org

This page belongs to course 2WF80 - Introduction to cryptology. This course is offered at TU/e as part of the bachelor's elective package 'Security'. The official page is here.

Contents
Classical systems (Caesar cipher, Vigenère, Playfair, rotor machines), shift register sequences, DES, RC4, RSA, Diffie-Hellman key exchange, cryptanalysis by using statistics, factorization, attacks on WEP (aircrack).

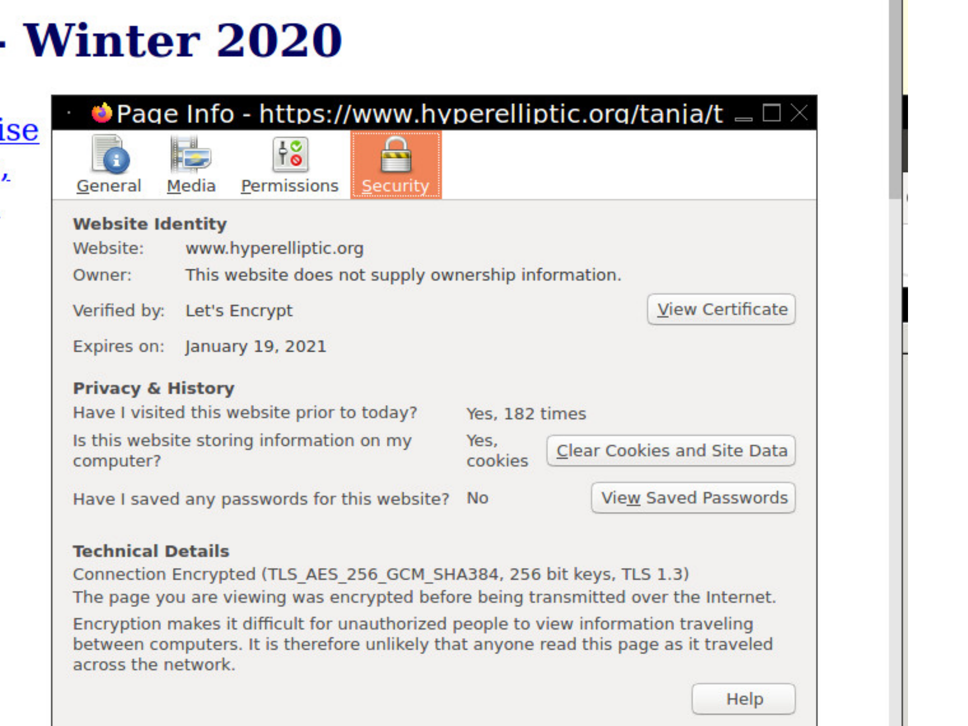
Some words up front: Crypto is an exciting area of research. Learning crypto makes you more aware of the limitations of security and privacy which might make you feel *less* secure but that's just a more accurate impression of reality and it a good step to improve your security.
Here is a nice link collection of software to help you stay secure https://prism-break.org/en/ and private https://www.privacytools.io/.

Announcements

Website Identity
Website: www.hyperelliptic.org
Owner: This website does not supply ownership information.
Verified by: Let's Encrypt
Expires on: January 19, 2021

Privacy & History
Have I visited this website prior to today? Yes, 182 times
Is this website storing information on my computer? Yes, cookies
Have I saved any passwords for this website? No

Technical Details
Connection Encrypted (TLS_AES_256_GCM_SHA384, 256 bit keys, TLS 1.3)
The page you are viewing was encrypted before being transmitted over the Internet.
Encryption makes it difficult for unauthorized people to view information traveling between computers. It is therefore unlikely that anyone read this page as it traveled across the network.



- Winter 2020

ise

Page Info - https://www.hyperelliptic.org/tania/t

Website Identity

Website: www.hyperelliptic.org

Owner: This website does not supply ownership information.

Verified by: Let's Encrypt

Expires on: January 19, 2021

Privacy & History

Have I visited this website prior to today? Yes, 182 times

Is this website storing information on my computer? Yes, cookies

Have I saved any passwords for this website? No

Technical Details

Connection Encrypted (TLS_AES_256_GCM_SHA384, 256 bit keys, TLS 1.3)

The page you are viewing was encrypted before being transmitted over the Internet.

Encryption makes it difficult for unauthorized people to view information traveling between computers. It is therefore unlikely that anyone read this page as it traveled across the network.