IT-Security 1

Chapter 13: Off-the-Record Messaging

Prof. Dr.-Ing. Ulrike Meyer

WS 15/16

Motivation

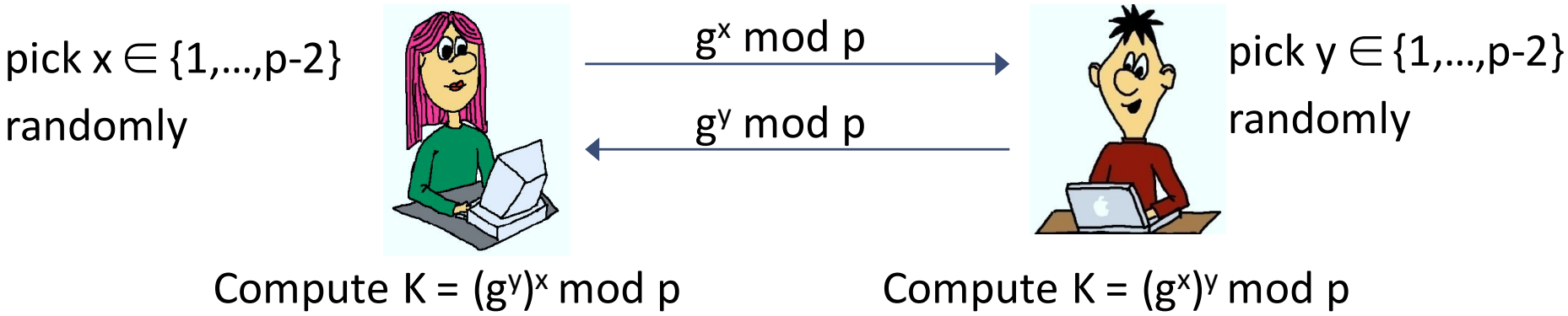
- PGP and S/MIME well suited to protect email communication in business contexts
  - Offer end-to-end confidentiality, data authenticity, non-repudiation
- Much of our social communication and sometimes even business communication is meant to be off-the-record
  - i.e. If Alice and Bob are communicating, Alice should not be able to proof to a third person what Bob has written or said
- Such off-the-record communication cannot be realized with the help of PGP, S/MIME
  - PGP and S/MIME use signatures to authenticate messages
  - Signatures make the messages non-repudiable

Security Requirements for Off-the-Record

- End-to-end confidentiality
  - i.e. only the two communicating entities are able to obtain the plaintext of the messages exchanged between them
- Perfect forward secrecy
  - i.e. even if (long-term) keys of the entities are compromised in the future, the past communication remains confidential
- Data authenticity
  - i.e. each entity is ensured that the messages indeed originate from the desired other entity
- Repudiation = Plausible Deniability
  - i.e. each of the two parties can later on convincingly deny that he has sent a particular plaintext message, i.e. none of the parties can proof to any third party, that the other party has sent a particular message

Cryptographic Primitives Used: Perfect Forward Secrecy

- Make use of short-lived encryption keys that are generated as needed and discarded immediately after use
- Ensure that it is impossible to rederive the keys used from long-lived keying material
- Uses Diffie-Hellman to generate short lived keys: p prime, g generator of Z_p^*



- Delete private exponent and DH key after use

Cryptographic Primitives Used: Digital Signatures

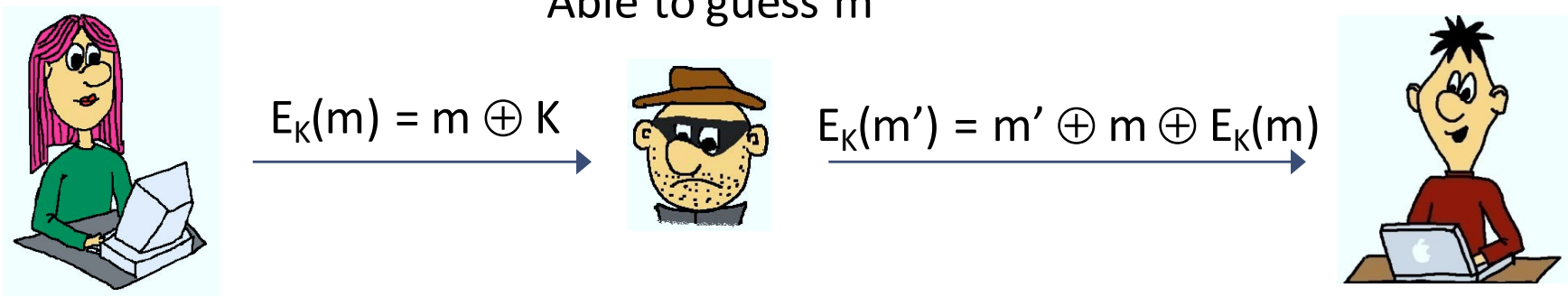
- Digital signatures are used to authenticate Alice and Bob in the first Diffie Hellman key exchange
- Digital signatures are not used to sign any other messages exchanged
- Signature keys are long-lived keys, Alice and Bob need to obtain an authentic copy of each other's signature keys

Cryptographic Primitives Used: MACs and repudiablity

- Messages exchanged between Alice and Bob are protected with a MAC (e.g. HMAC)
- The MAC ensures Alice that messages are indeed generated by Bob (and vice versa)
- However, Alice is not able to proof to anyone else, that Bob has generated a specific message, as she could have generated the message herself (and vice versa)
- Even more: after Alice has checked the authenticity of all messages protected by Bob with a specific MAC key, this key is published such that anyone can now generate messages protected with this key

Cryptographic Primitives Used: Malleable encryption

- In order to ensure that messages exchanged can indeed be forged by anyone, malleable encryption is used
- I.e. an encryption scheme that allows anyone to change a known plaintext of a given ciphertext to another meaningful plaintext without knowledge of the key
- Examples for malleable encryption schemes are stream ciphers, e.g. AES in counter mode



- Note that Alice sill uses MAC to proof authenticity of m to Bob

Putting it all together: OTR v1 (insecure)

p prime, g generator of Z_p^*, E: AES in counter mode, H: cryptographic hash function,
s_A: Alice's private signature key, p_B: Bob's public signature verification key

s_A, p_B

s_B, p_A

$x_1 \in_R \{1,\ldots,p-2\}$

$\text{sig}_{s_A}(g^{x_1})$

$\text{sig}_{s_B}(g^{y_1})$

$y_1 \in_R \{1,\ldots,p-2\}$

$K_{11} = H(g^{y_1 x_1})$

$x_2 \in_R \{1,\ldots,p-2\}$

$g^{x_2}, E_{K11}(M_1), \text{MAC}_{H(K_{11})}(g^{x_2}, E_{K11}(M_1))$

$K_{21} = H(g^{x2y1})$

$y_2 \in_R \{1,\ldots,p-2\}$

$g^{y_2}, E_{K21}(M_2), \text{MAC}_{H(K_{21})}(g^{y_2}, E_{K21}(M_2))$

Delete x_1

$K_{22} = H(g^{y_2 x_2})$

$x_3 \in_R \{1,\ldots,p-2\}$

$g^{x_3}, E_{K22}(M_3), \text{MAC}_{H(K_{22})}(g^{x_3}, E_{K22}(M_3))$

Delete y_1

$K_{32} = H(g^{x_3 y_2})$

$y_3 \in_R \{1,\ldots,p-2\}$

$g^{y_3}, E_{K32}(M_4), \text{MAC}_{H(K_{32})}(g^{y_3}, E_{K32}(M_4))$

Delete x_2

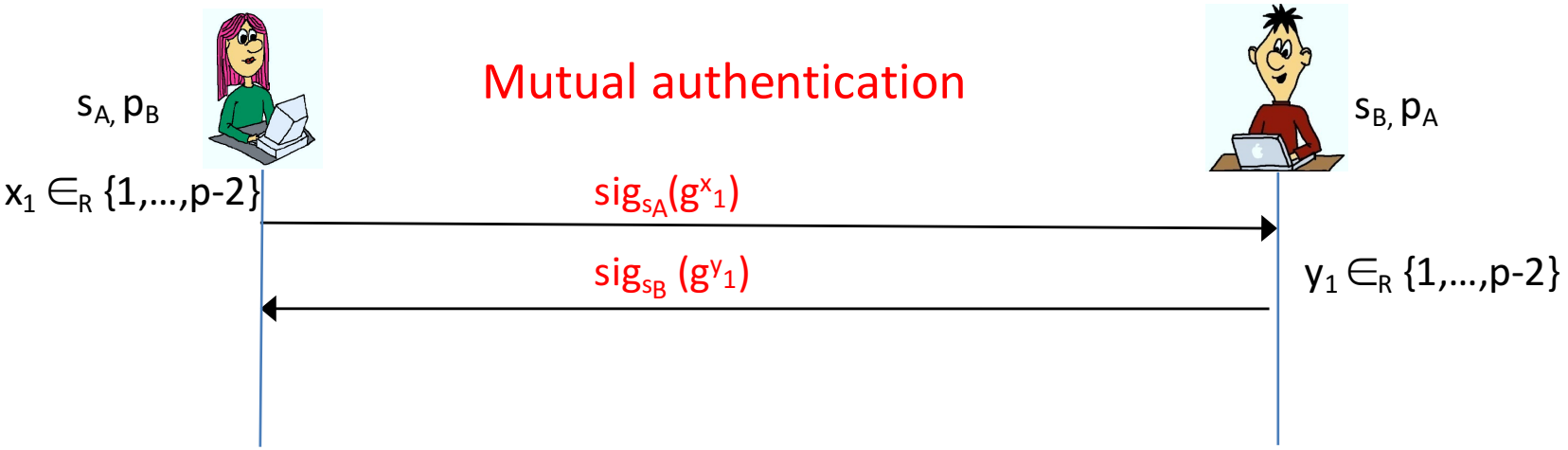
$K_{33} = H(g^{y_3 x_3})$

$x_4 \in_R \{1,\ldots,p-2\}$

$g^{x_4}, E_{K33}(M_5), \text{MAC}_{H(K_{33})}(g^{x_4}, E_{K33}(M_5)), H(K_{11})$

Authentication in OTR v1

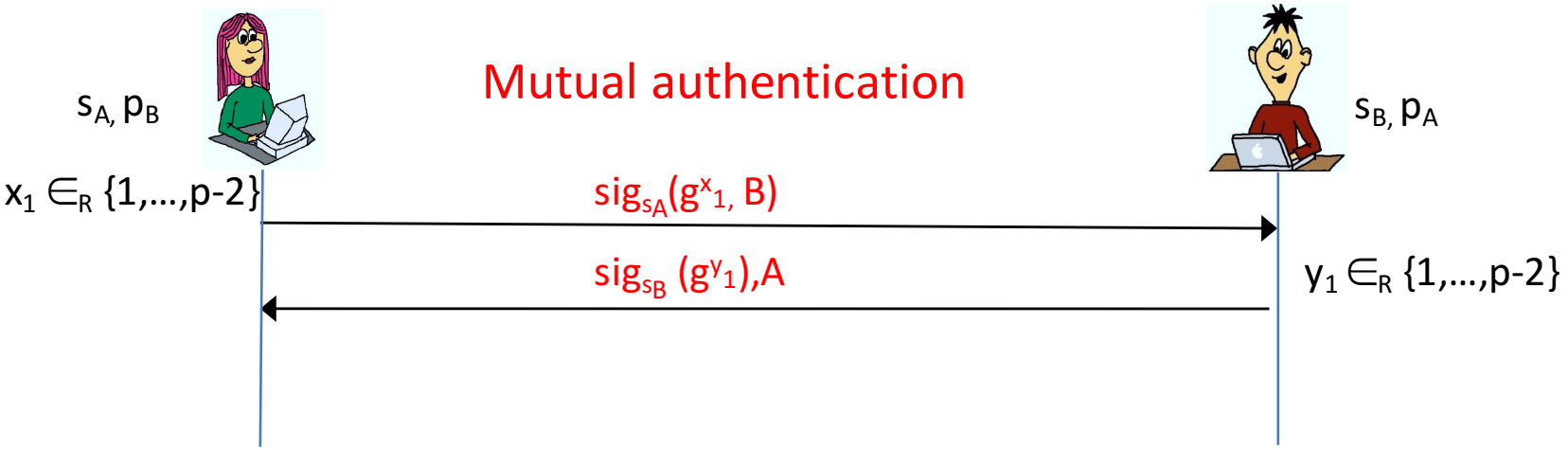
p prime, g generator of Z_p^*, E: AES in counter mode, H: cryptographic hash function,
s_A: Alice's private signature key, p_B: Bob's public signature verification key



- Problem with this way to authenticate the DH-Key Agreement?
  - No binding to identifiers, i.e. Eve can make Alice believe she communicates to Bob and make Bob believe he communicates with her
    - May e.g. lead to Bob bashing about Alice, assuming he speaks OTR to Eve
  - No freshness guarantees , i.e. a single compromised x1 allows attacker to impersonate Alice to anyone
- Sounds familiar? Check out Chapter 5 again

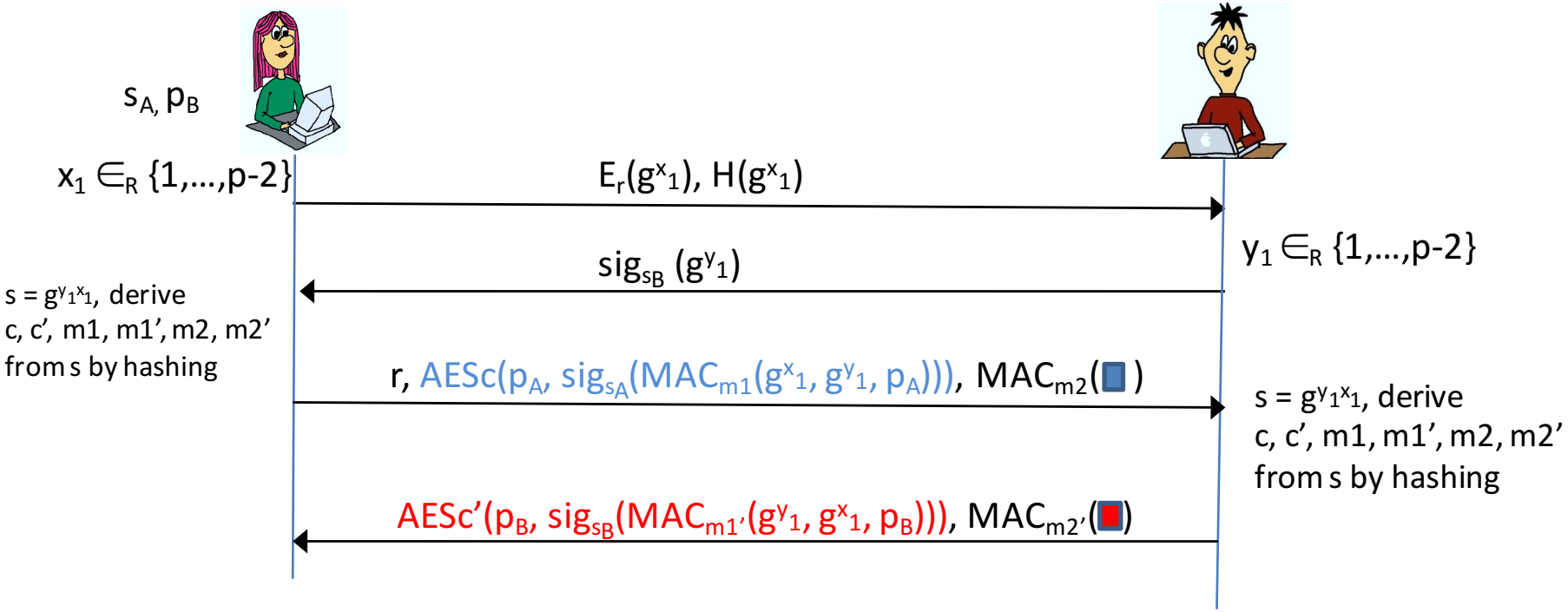
Quick fix?

p prime, g generator of Z_p^*, E: AES in counter mode, H: cryptographic hash function, s_A: Alice's private signature key, p_B: Bob's public signature verification key



- Fixing the first problem by including identifiers?
  - Violates deniability! Alice can no longer deny that she communicated with Bob
  - Even though what she said may stay confidential

Better Fix: Use e.g. SIGMA Protocol for Authentication



- Provides for mutual authentication
- Is deniable as no identifying information of the communication partner is signed

OTR v2 (secure), somewhat simplified here

p prime, g generator of Z_p^*, E: AES in counter mode, H: cryptographic hash function,
s_A: Alice's private signature key, p_B: Bob's public signature verification key

s_A, p_B

s_B, p_A

$x_1 \in_R \{1,\ldots,p-2\}$

Run the SIGMA protocol above

$y_1 \in_R \{1,\ldots,p-2\}$

$K_{11} = H(g^{y_1 x_1})$

$x_2 \in_R \{1,\ldots,p-2\}$

$g^{x_2}, E_{K11}(M_1), MAC_{H(K_{11})}(g^{x_2}, E_{K11}(M_1))$

$K_{21} = H(g^{x2y1})$

$y_2 \in_R \{1,\ldots,p-2\}$

$g^{y_2}, E_{K21}(M_2), MAC_{H(K_{21})}(g^{y_2}, E_{K21}(M_2))$

Delete x_1,

$K_{22} = H(g^{y_2 x_2})$

$x_3 \in_R \{1,\ldots,p-2\}$

$g^{x_3}, E_{K22}(M_3), MAC_{H(K_{22})}(g^{x_3}, E_{K22}(M_3))$

Delete y_1,

$K_{32} = H(g^{x_3 y_2})$

$y_3 \in_R \{1,\ldots,p-2\}$

$g^{y_3}, E_{K32}(M_4), MAC_{H(K_{32})}(g^{y_3}, E_{K32}(M_4))$

Delete x_2,

$K_{33} = H(g^{y_3 x_3})$

$x_4 \in_R \{1,\ldots,p-2\}$

$g^{x_4}, E_{K33}(M_5), MAC_{H(K_{33})}(g^{x_4}, E_{K33}(M_5)), H(K_{11})$

Properties of OTR v2

p prime, g generator of Z_p^*, E: AES in counter mode, H: cryptographic hash function,
s_A: Alice's private signature key, p_B: Bob's public signature verification key

s_A, p_B

Confidentiality

s_B, p_A

$x_1 \in_R \{1,\ldots,p-2\}$

Run the SIGMA protocol above

$y_1 \in_R \{1,\ldots,p-2\}$

$K_{11} = H(g^{y_1 x_1})$

$x_2 \in_R \{1,\ldots,p-2\}$

$g^{x_2}, E_{K11}(M_1), MAC_{H(K_{11})}(g^{x_2}, E_{K11}(M_1))$ →

$K_{21} = H(g^{x2y1})$

$y_2 \in_R \{1,\ldots,p-2\}$

← $g^{y_2}, E_{K21}(M_2), MAC_{H(K_{21})}(g^{y_2}, E_{K21}(M_2))$

Delete x_1,

$K_{22} = H(g^{y_2 x_2})$

$x_3 \in_R \{1,\ldots,p-2\}$

$g^{x_3}, E_{K22}(M_3), MAC_{H(K_{22})}(g^{x_3}, E_{K22}(M_3))$ →

Delete y_1,

$K_{32} = H(g^{x_3 y_2})$

$y_3 \in_R \{1,\ldots,p-2\}$

← $g^{y_3}, E_{K32}(M_4), MAC_{H(K_{32})}(g^{y_3}, E_{K32}(M_4))$

Delete x_2,

$K_{33} = H(g^{y_3 x_3})$

$x_4 \in_R \{1,\ldots,p-2\}$

$g^{x_4}, E_{K33}(M_5), MAC_{H(K_{33})}(g^{x_4}, E_{K33}(M_5)), H(K_{11})$ →

Properties of OTR v2

p prime, g generator of Z_p^*, E: AES in counter mode, H: cryptographic hash function,
s_A: Alice's private signature key, p_B: Bob's public signature verification key

s_A, p_B

Perfect forward secrecy

s_B, p_A

$x_1 \in_R \{1,\ldots,p-2\}$

Run the SIGMA protocol above

$y_1 \in_R \{1,\ldots,p-2\}$

$K_{11} = H(g^{y_1 x_1})$

$x_2 \in_R \{1,\ldots,p-2\}$

$g^{x_2}, E_{K11}(M_1), MAC_{H(K_{11})}(g^{x_2}, E_{K11}(M_1))$ →

$K_{21} = H(g^{x2y1})$

$y_2 \in_R \{1,\ldots,p-2\}$

← $g^{y_2}, E_{K21}(M_2), MAC_{H(K_{21})}(g^{y_2}, E_{K21}(M_2))$

Delete x_1

$K_{22} = H(g^{y_2 x_2})$

$x_3 \in_R \{1,\ldots,p-2\}$

$g^{x_3}, E_{K22}(M_3), MAC_{H(K_{22})}(g^{x_3}, E_{K22}(M_3))$ →

Delete y_1,

$K_{32} = H(g^{x,_3 y_2})$

$y_3 \in_R \{1,\ldots,p-2\}$

← $g^{y_3}, E_{K32}(M_4), MAC_{H(K_{32})}(g^{y_3}, E_{K32}(M_4))$

Delete x_2

$K_{33} = H(g^{y_3 x_3})$

$x_4 \in_R \{1,\ldots,p-2\}$

$g^{x_4}, E_{K33}(M_5), MAC_{H(K_{33})}(g^{x_4}, E_{K33}(M_5)), H(K_{11})$ →

Properties of OTR v2

p prime, g generator of Z_p^*, E: AES in counter mode, H: cryptographic hash function,
s_A: Alice's private signature key, p_B: Bob's public signature verification key

s_A, p_B

Integrity and authenticity of messages

s_B, p_A

$x_1 \in_R \{1,\ldots,p-2\}$

Run the SIGMA protocol above

$y_1 \in_R \{1,\ldots,p-2\}$

$K_{11} = H(g^{y_1 x_1})$

$x_2 \in_R \{1,\ldots,p-2\}$

$g^{x_2}, E_{K11}(M_1), MAC_{H(K_{11})}(g^{x_2}, E_{K11}(M_1))$ →

$K_{21} = H(g^{x2y1})$

$y_2 \in_R \{1,\ldots,p-2\}$

← $g^{y_2}, E_{K21}(M_2), MAC_{H(K_{21})}(g^{y_2}, E_{K21}(M_2))$

Delete x_1

$K_{22} = H(g^{y_2 x_2})$

$x_3 \in_R \{1,\ldots,p-2\}$

$g^{x_3}, E_{K22}(M_3), MAC_{H(K_{22})}(g^{x_3}, E_{K22}(M_3))$ →

Delete y_1,

$K_{32} = H(g^{x,_3 y_2})$

$y_3 \in_R \{1,\ldots,p-2\}$

← $g^{y_3}, E_{K32}(M_4), MAC_{H(K_{32})}(g^{y_3}, E_{K32}(M_4))$

Delete x_2

$K_{33} = H(g^{y_3 x_3})$

$x_4 \in_R \{1,\ldots,p-2\}$

$g^{x_4}, E_{K33}(M_5), MAC_{H(K_{33})}(g^{x_4}, E_{K33}(M_5)), H(K_{11})$ →

Properties of OTR v2

p prime, g generator of Z_p^*, E: AES in counter mode, H: cryptographic hash function,
s_A: Alice's private signature key, p_B: Bob's public signature verification key

s_A, p_B

Integrity and authenticity of messages

s_B, p_A

$x_1 \in_R \{1,\ldots,p-2\}$

Run the SIGMA protocol above

$y_1 \in_R \{1,\ldots,p-2\}$

$K_{11} = H(g^{y_1 x_1})$

$x_2 \in_R \{1,\ldots,p-2\}$

$g^{x_2}, E_{K11}(M_1), MAC_{H(K_{11})}(g^{x_2}, E_{K11}(M_1))$ →

$K_{21} = H(g^{x2y1})$

$y_2 \in_R \{1,\ldots,p-2\}$

← $g^{y_2}, E_{K21}(M_2), MAC_{H(K_{21})}(g^{y_2}, E_{K21}(M_2))$

Delete x_1

$K_{22} = H(g^{y_2 x_2})$

$x_3 \in_R \{1,\ldots,p-2\}$

$g^{x_3}, E_{K22}(M_3), MAC_{H(K_{22})}(g^{x_3}, E_{K22}(M_3))$ →

Delete y_1,

$K_{32} = H(g^{x,_3 y_2})$

$y_3 \in_R \{1,\ldots,p-2\}$

← $g^{y_3}, E_{K32}(M_4), MAC_{H(K_{32})}(g^{y_3}, E_{K32}(M_4))$

Delete x_2

$K_{33} = H(g^{y_3 x_3})$

$x_4 \in_R \{1,\ldots,p-2\}$

$g^{x_4}, E_{K33}(M_5), MAC_{H(K_{33})}(g^{x_4}, E_{K33}(M_5)), H(K_{11})$ →

Properties of OTR v2

p prime, g generator of Z_p^*, E: AES in counter mode, H: cryptographic hash function,
s_A: Alice's private signature key, p_B: Bob's public signature verification key

s_A, p_B

Deniability

s_B, p_A

$x_1 \in_R \{1,\ldots,p-2\}$

Run the SIGMA protocol above

$y_1 \in_R \{1,\ldots,p-2\}$

$K_{11} = H(g^{y_1 x_1})$

$x_2 \in_R \{1,\ldots,p-2\}$

$g^{x_2}, E_{K11}(M_1), MAC_{H(K_{11})}(g^{x_2}, E_{K11}(M_1))$ →

$K_{21} = H(g^{x2y1})$

$y_2 \in_R \{1,\ldots,p-2\}$

← $g^{y_2}, E_{K21}(M_2), MAC_{H(K_{21})}(g^{y_2}, E_{K21}(M_2))$

Delete x_1

$K_{22} = H(g^{y_2 x_2})$

$x_3 \in_R \{1,\ldots,p-2\}$

$g^{x_3}, E_{K22}(M_3), MAC_{H(K_{22})}(g^{x_3}, E_{K22}(M_3))$ →

Delete y_1,

$K_{32} = H(g^{x,_3 y_2})$

$y_3 \in_R \{1,\ldots,p-2\}$

← $g^{y_3}, E_{K32}(M_4), MAC_{H(K_{32})}(g^{y_3}, E_{K32}(M_4))$

Delete x_2

$K_{33} = H(g^{y_3 x_3})$

$x_4 \in_R \{1,\ldots,p-2\}$

$g^{x_4}, E_{K33}(M_5), MAC_{H(K_{33})}(g^{x_4}, E_{K33}(M_5)), H(K_{11})$ →

Further Reading and Resources

1. Nikita Borisov, Ian Goldberg, Eric Brewer: *Off-the-record Communication, or, Why Not To Use PGP,* ACM WPES 2004
2. Mario Raimondo, Rosario Gennaro, Hugo Krawczyk: *Secure Off-the-Record Messaging*, ACM WPES 2005
3. OTR protocol version 2, fixes the flaws of 1. described in 2. https://otr.cypherpunks.ca/Protocol-v2-3.1.0.html
4. Chris Alexander and Ian Goldberg: Improved User Authentication in OTR, ACM WPES 2007
   - Solves the problem that users do typically not yet have an authentic copy of each other's public key available