Vorlesung Kryptographie

Aufzeichnungen vom 17.5.

3.3 EL-Gamal-Verschlüsselung (17.5.17)

D-H-Variante: a wird nur einmal erzeugt, b für jede Nachricht erneut.

EL-Gamal-V.: VORAB
- Veröffentliche Primzahl p und $1 < g < p$
- Alice wählt a (geheim) und veröffentlicht $g^a \bmod p$
- Bob wählt zufällig b und berechnet $g^b \bmod p$ und $K := (g^a)^b \bmod p$

- Bob verschlüsselt Nachricht m als $c \equiv m \cdot K \bmod p$; und sendet c und $g^b \bmod p$ an Alice.
- Alice berechnet $K = (g^b)^a \bmod$ und
- $m \equiv c \cdot K^{-1} \bmod p \equiv c \cdot (g^b)^{p-1-a}$

Korrekt: $c \cdot K^{-1} \equiv m \cdot K \cdot K^{-1} \equiv m \cdot 1 \equiv m \bmod p$

bzw $\left(g^{b(p-1)} \cdot g^{b(-a)}\right) \equiv \left(g^{\varphi(p)}\right)^b \cdot g^{-ab}$

$K \equiv g^{ab}$; $K^{-1} \equiv g^{-ab}$ ✓

$\equiv 1$ (Euler-Fermat)

machbar: Genauso wie D-H ✓

Sicher: Eve kennt g^a, g^b, $m \cdot k \bmod p$ (= g^{ab})

aber nicht k (= g^{ab}), a.

So sicher wie D-H bzw. Zufallszahl.

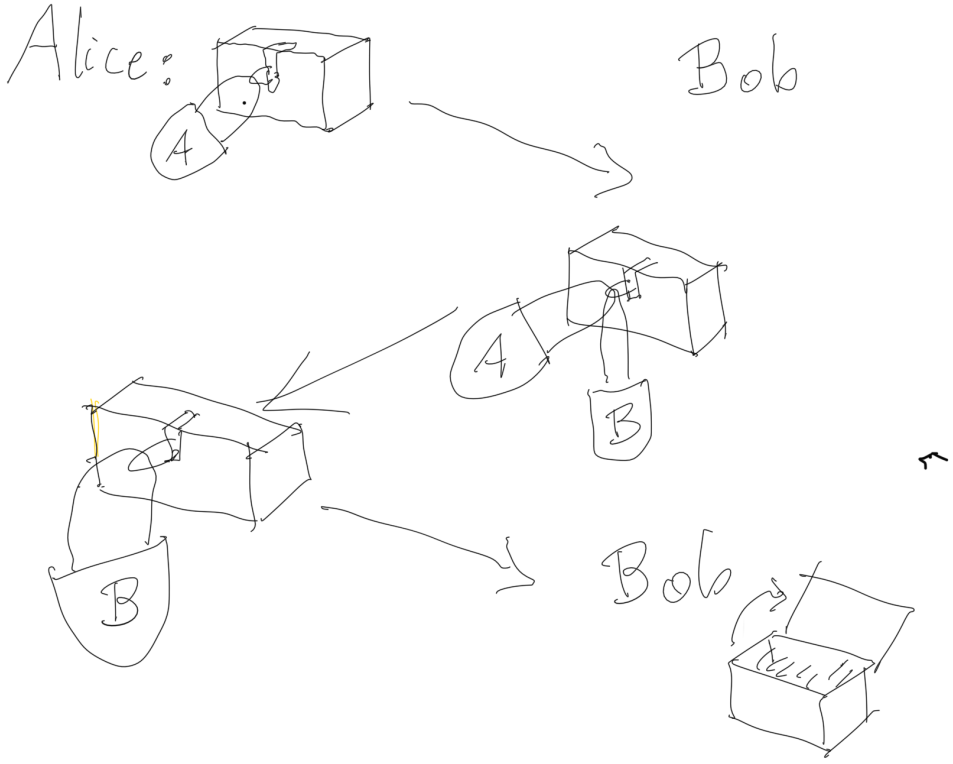
Bemerk:
- Anders als bei RSA können alle dasselbe p benutzen
- Wegen der zufälligen Wahl von b würden beim zweimaligen Verschlüsseln von m zwei verschiedene Geheimtexte c_1, c_2 erzeugt.

- Das hier ist die klassische ElGamal-Methode. Hier war $f(k,m) \equiv m \cdot k \bmod p$; also entschlüsseln: $f^*(k,m) \equiv m \cdot k^{-1} \bmod p$.
f ist symm. Verfahren!
Statt diesem speziellen f kann irgendein symm. Verfahren benutzt werden (z.B. one-time-pad; AES) Oder:

3.4 Shamirs No-Key-Protokoll

Ohne (private) Schlüssel, aber mit öffentlicher Primzahl p.

Idee:



Shamirs No-Key Protokoll:

VORAB (geheim):
- Alice & Bob einigen sich auf Primzahl p
- Alice erzeugt a, a' mit $a \cdot a' \equiv 1 \bmod p-1$
- Bob " b, b' " $b \cdot b' \equiv 1 \bmod p-1$

- Alice schickt $m^a \bmod p$ an Bob
- Bob schickt $m^{ab} \bmod p$ an Alice
- Alice " $m^{aba'} \bmod p$ an Bob
- Bob berechnet $m^{aba'b'} \equiv m \bmod p$

($\varphi(p) = p-1$)

Korrekt:

$$m^{aba'b'} \equiv (m^{aa'})^{bb'} \equiv (m^{k \cdot \varphi(p)+1})^{bb'} \bmod p$$

$$\equiv ((m^{\varphi(p)})^k \cdot m)^{bb'} \equiv m^{bb'} \equiv \underbrace{m^{l \cdot \varphi(p)+1}}_{m^{\varphi(p)} \equiv 1} \equiv m \bmod p$$

$\equiv 1^k$

(Euler-Fermat)

Hier: $ggT(m,p)=1$ garantiert $(1 < m < p)$ (↑ Primzahl)

machbar: ✓ (Potenzieren mod m, a' berechnen (mit erweitertem Eukl. Algor.))

sicher: (höchstens) so sicher wie dlog mod p (Üb 15 Blatt 4)

Bemerk. Das ist ein symmetrisches Verfahren, denn Verschlüsseln: $m^a \bmod p = f(a,m)$

Entschlüsseln: $c^{a'} \bmod p = f^*(a',c)$

also $f = f^*$; und a' kann leicht aus a berechnet werden (u.U.)

Außerdem: $f(a, f(b,m)) \equiv (m^b)^a \bmod p$

$\equiv (m^a)^b \equiv f(b, f(a,m))$

also auch in diesem Sinne symmetrisch! (wichtig für No-Key-Verfahren)

- Da dieses Verfahren symmetrisch, kann es auch als f im El-Gamal-Verschlüsselungs-verfahren dienen.

Später wichtig: Signaturen. Alles oben ist anfällig für (wo)man-in-the-middle-attack

Alice ⟷ Eve ⟷ Bob

Falls also Eve sich Alice gegenüber als Bob ausgibt u.u.
Wünschenswert: Sichere Authentifizierung
("Ich bin wirklich Bob")

Eigenschaften:
- Identitätseigsch.
- Echtheitseigsch: „Nachricht kommt wirklich von Bob"
- Originaleigsch: Nachricht wurde nicht nach dem Signieren verändert
- Verifikationseigsch: Jeder kann die Echtheit der Nachricht prüfen.

Einfache Grundidee: Alice schickt $c = f(e_B, m)$
Bob kennt nun m. " " auch $c' = f(d_A, m)$ an Bob.

Bobs public key

Alices private key

Bob entschlüsselt c' mit Alices öffentlichem Schlüssel als m' und checkt ob $m = m'$.
Falls ja: nur Alice kann c' gesendet haben

Identität: ✓ Echtheit ✓ Original ✓
Verifizieren ✓

Nachteil: offenbar: Jetzt kann jeder m lesen: c' mit e_A entschlüsseln.
Lösung: Statt m benutze „Fingerabdruck": Hashfunktion.