

Autentizační kód zprávy

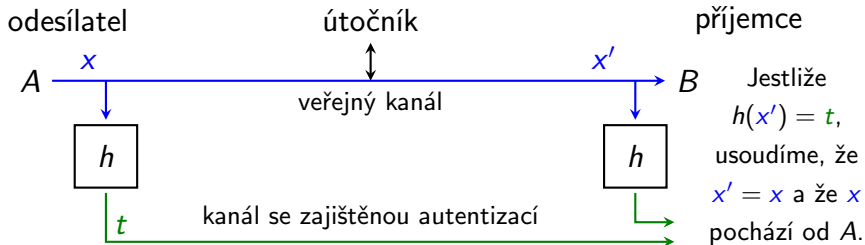
Andrew Kozlík

KA MFF UK

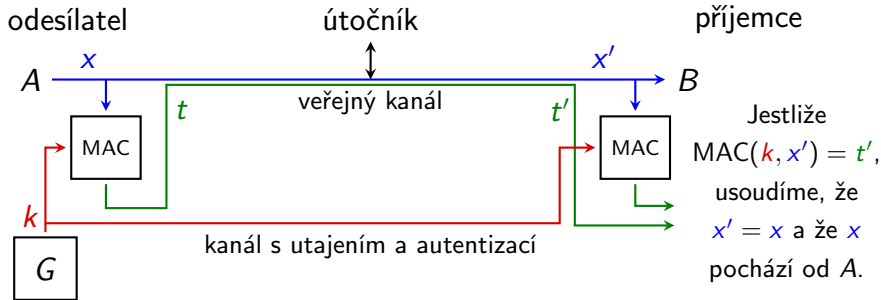
Autentizační kód zprávy

- ▶ Anglicky: message authentication code (MAC).
- ▶ MAC algoritmus je v podstatě hashovací funkce s klíčem:
$$\text{MAC} : \{0, 1\}^k \times \{0, 1\}^* \rightarrow \{0, 1\}^n, \quad \text{kde } k, n \in \mathbb{N}.$$
- ▶ MAC slouží k autentizaci původce zprávy.
- ▶ Implicitně zahrnuje i zajištění integrity zprávy.
- ▶ Obraz nazýváme „tag“ nebo prostě „MAC“.
- ▶ Tag můžeme připojit ke zprávě jako jakýsi „podpis“. Není to podpis, protože nezajišťuje nepopiratelnost!

Zajištění autentizace pomocí hashovací funkce



Zajištění autentizace pomocí MACu



Typy útoků na MAC

- ▶ **Known message útok**

Útočník zná jeden nebo více párů (x_i, t_i) , kde $t_i = \text{MAC}(k, x_i)$ a k je konstantní.

- ▶ **Chosen message útok**

Útočník volí zprávy x_1, \dots, x_n a dozvídá se hodnoty $t_i = \text{MAC}(k, x_i)$, kde k je konstantní.

- ▶ Cílem je vytvořit novou dvojici (x, t) , kde $t = \text{MAC}(k, x)$.

Šifra nezajišťuje ani integritu ani autentizaci

- ▶ Alice posílá do banky platební příkaz pro Boba.
- ▶ Příkaz má standardní formát

$$x = \text{pořadové číslo} \parallel N_A \parallel N_B \parallel \text{částka}$$

a zašifruje se proudovou šifrou

$$y = x \oplus s,$$

kde s je keystream, N_A a N_B jsou čísla účtů Alice a Boba.

- ▶ Cestou do banky je zašifrovaný příkaz zadržen Evou.
- ▶ Ta ho pozmění a pošle dál:

$$y' = y \oplus (0 \dots 0 \parallel 0 \dots 0 \parallel N_B \oplus N_E \parallel 0 \dots 0).$$

- ▶ Po dešifrování y' v bance dostaneme

$$\text{pořadové číslo} \parallel N_A \parallel N_E \parallel \text{částka}.$$

Autentizované šifrování

- ▶ Encrypt-then-MAC
 - ▶ Máme klíče k_1 a k_2 (jeden může být odvozený z druhého).
 - ▶ Spočítáme $y = E(k_1, x)$ a $t = \text{MAC}(k_2, y)$.
 - ▶ Pošleme (y, t) .
- ▶ Existují operační režimy blokových šifer, které kombinují utajení a autentizaci.
 - ▶ Výstupem je šifrový text a tag.
 - ▶ Například: Galois Counter Mode (GCM).
- ▶ Obrana proti replay útokům:
Do zprávy přidáme důkaz čerstvosti, např. pořadové číslo.

Konstrukce MAC algoritmu z hashovací funkce

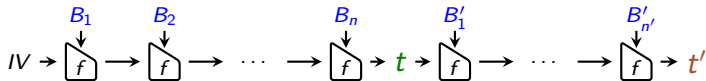
► **Nápad:**

$$\text{MAC}(k, x) := h(k \parallel x).$$

► **Problém:**

Když h má Merkleovu-Damgårdovu konstrukci:

- Útočník odposlechne (x, t) , kde $t = h(k \parallel x)$.
- Pro libovolné x' může spočítat $h(k \parallel \bar{x} \parallel x')$ bez znalosti k .
- Stačí navázat na posloupnost volání kompresní funkce:



- Tím získá tag t' pro zprávu $\bar{x} \parallel x'$, aniž by znal klíč k .

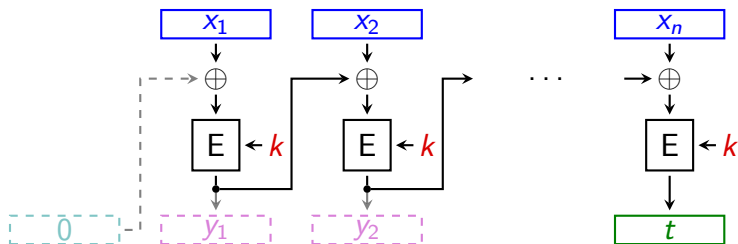
► **Řešení:**

$$\text{HMAC}(k, x) := h\left(\left(k \oplus \text{opad}\right) \parallel h\left(\left(k \oplus \text{ipad}\right) \parallel x\right)\right),$$

kde **opad** a **ipad** jsou standardem dané konstanty.

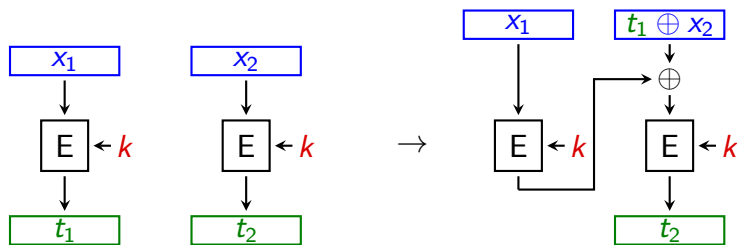
Konstrukce MAC algoritmu z blokové šifry

- ▶ **Nápad:** CBC-MAC
 - ▶ Zprávu vyplníme a rozdělíme na bloky x_1, \dots, x_n .
 - ▶ „Zašifrujeme“ blokovou šifrou v CBC režimu, $IV = 0$.
 - ▶ Jako tag použijeme pouze poslední blok výstupu.
- ▶ Pokud zprávu zároveň šifrujeme, tak pro šifrování a CBC-MAC je obecně nutno použít různé klíče!



Konstrukce MAC algoritmu z blokové šifry

- ▶ **Problém:** Known message útok na CBC-MAC.
 - ▶ Útočník odposlechne (x_1, t_1) a (x_2, t_2) , kde $t_i = \text{CBC-MAC}(k, x_i)$.
 - ▶ Pro jednoduchost předpokládejme, že délka x_i je rovna délce jednoho bloku šifry, potom $t_i = E(k, x_i)$.
 - ▶ Označme $x_3 := t_1 \oplus x_2$.
 - ▶ Potom $\text{CBC-MAC}(k, x_1 \parallel x_3) = E(k, E(k, x_1) \oplus x_3) = E(k, t_1 \oplus t_1 \oplus x_2) = t_2$.
 - ▶ Sestrojili jsme novou platnou dvojici $(x_1 \parallel (t_1 \oplus x_2), t_2)$.



Konstrukce MAC algoritmu z blokové šifry

► **Řešení:** EMAC

Výsledek CBC-MACu navíc zašifrujeme jiným klíčem:

$$\text{EMAC}(k_1, k_2, x) := E(k_2, \text{CBC-MAC}(k_1, x))$$

► **Problém:** Chosen message útok

► Podaří-li se útočníkovi narazit na kolizi (x_1, t) a (x_2, t) , kde $t = \text{EMAC}(k_1, k_2, x_1) = \text{EMAC}(k_1, k_2, x_2)$, pak $\text{CBC-MAC}(k_1, x_1) = \text{CBC-MAC}(k_1, x_2)$.

► Necht' x' je libovolný řetězec o délce jednoho bloku.

► Uvažujeme-li chosen message útok, pak se útočník může dotázat na $\text{EMAC}(k_1, k_2, (x_1 \parallel x')) = t_3$.

Chosen message útok na EMAC

- ▶ Pokračování:

- ▶ Nyní platí:

$$\begin{aligned} \text{EMAC}(k_1, k_2, (x_2 \parallel x')) &= E(k_2, \text{CBC-MAC}(k_1, (x_2 \parallel x'))) \\ &= E(k_2, E(k_1, \text{CBC-MAC}(k_1, x_2) \oplus x')) \\ &= E(k_2, E(k_1, \text{CBC-MAC}(k_1, x_1) \oplus x')) \\ &= E(k_2, \text{CBC-MAC}(k_1, (x_1 \parallel x'))) \\ &= \text{EMAC}(k_1, k_2, (x_1 \parallel x')) = t_3. \end{aligned}$$

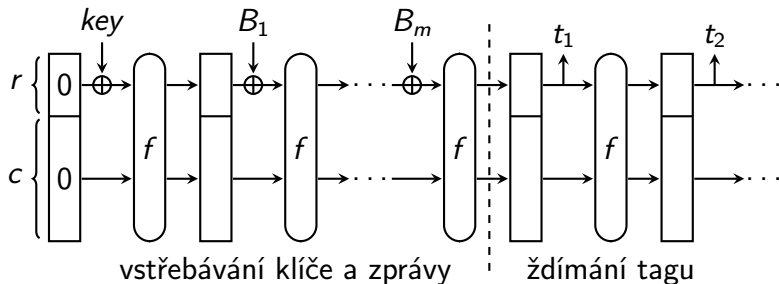
- ▶ Sestrojili jsme novou platnou dvojici $((x_2 \parallel x'), t_3)$, ačkoliv se útočník na tag pro $(x_2 \parallel x')$ nikdy nedotázal.

- ▶ **Řešení:**

1. Omezíme počet zpráv, pro které se k_1 a k_2 použijí, aby kolize tagů byly málo pravděpodobné.
2. Jako tag použijeme jen několik prvních bitů výstupu EMACu, aby útočník nemohl z tagu poznat kolize CBC-MACu.

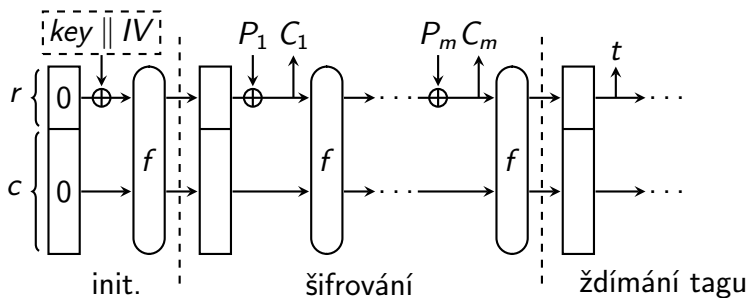
Konstrukce MAC algoritmu z houby

- ▶ Hashovací funkci, která je založena na houbovitě konstrukci lze použít jako MAC napřímo: $h(\text{key} \parallel \text{message})$.
 - ▶ Merkleova-Damgårdova konstrukce:
Výstupem je celý vnitřní stav.
 - ▶ Houbovitá konstrukce:
Na výstupu chybí minimálně c bitů stavu houby.
- ▶ Na tomto principu funguje KMAC (KECCAK Message Authentication Code algorithm).



Autentizované šifrování duplexní konstrukcí

- ▶ Otevřený text se rozdělí na bloky P_1, \dots, P_{m-1} délky $r - 3$ bitů a blok P_m délky nejvýše $r - 3$ bitů.
- ▶ K bloku P_m se připojí bitová výplň tvaru 010^*1 a k ostatním bitová výplň 111 , aby všechny bloky měly délku r .
- ▶ Z bloků C_1, \dots, C_m se odřezou bity na pozicích výplně.
- ▶ Výsledkem je šifrový text $C_1 \parallel \dots \parallel C_m$ a tag t .



K čemu je výplň v duplexní konstrukci?

- ▶ První bit výplně je tzv. *frame bit*, který je 0 v posledním bloku a 1 v ostatních.
- ▶ Kdyby se použila výplň bez frame bitu, nebo kdyby se nepoužila žádná výplň, tak by bylo možné provést known plaintext attack, kterým se zkrátí zašifrovaná zpráva a zfalšuje tag.
- ▶ Příklad útoku při absenci frame bitu:
 - ▶ Útočník zachytí $(IV, C_1 \parallel \dots \parallel C_m, t)$ a zná P_m .
 - ▶ Útočník odešle $(IV, C_1 \parallel \dots \parallel C_{m-1}, t')$, kde $t' = P_m \oplus C_m$.
 - ▶ Po dešifrování příjemce získá $P_1 \parallel \dots \parallel P_{m-1}$ a tag t' se příjemci jeví jako validní.
- ▶ Frame bit brání útočníkovi ve vytvoření validního tagu pro zkrácenou zprávu.