Elementi di Crittografia
Sicurezza: altre nozioni — Oracoli, PRF/PRP

Integrazione di: EC-SN-25.txt (slide) + Trascrizione lezione 7

9 ottobre 2025

Sommario

Questo documento integra le slide del modulo "Sicurezza: altre nozioni" con la spiegazione del docente. Tratta: nozioni di sicurezza per messaggi multipli, attacchi chosen-plaintext e modellazione con oracoli, definizioni IND-CPA (singola e multipla), funzioni e permutazioni pseudocasuali (PRF/PRP, PRP forte), esempi e relazioni tra PRG e PRF; include anche note su sicurezza concreta e casi storici.

Indice

1 Contenuti e obiettivi

Contenuto slide. *Contenuti:*

1. *Altre nozioni di sicurezza*
2. *Oracoli*
3. *Funzioni e permutazioni pseudocasuali*

Spiegazione del docente. Obiettivo: estendere le nozioni di indistinguibilità alla situazione di messaggi multipli e a modelli di attacco più forti (chosen-plaintext). Introdurre oracoli per formalizzare le capacità dell'avversario. Presentare PRF/PRP come strumenti per costruire schemi CPA-sicuri. Ripasso della tecnica di riduzione: se un avversario rompe la costruzione, allora si ottiene un distinguisher contro la primitiva di base (es. PRG).

2 Indistinguibilità rispetto a messaggi multipli (passive eavesdropper)

2.1 Esperimento e definizione

Contenuto slide. *Esperimento* $\mathsf{PrivK}^{\mathit{eav\text{-}mult}}_{\mathcal{A},\Pi}(n)$*:*

1. $\mathcal{A}(1^n)$ *emette due liste* $M_0 = (m_{0,1}, \dots, m_{0,t})$ *e* $M_1 = (m_{1,1}, \dots, m_{1,t})$ *con* $|m_{0,i}| = |m_{1,i}|$ *per ogni* i.
2. *La sfida:* Chall *sceglie* $b \leftarrow \{0,1\}$, $k \leftarrow \mathsf{Gen}(1^n)$ *e calcola* $c_i \leftarrow \mathsf{Enc}_k(m_{b,i})$ *per ogni* i.
3. $\mathcal{A}$ *riceve* $c = (c_1, \dots, c_t)$ *e risponde con* $b' \in \{0,1\}$.
4. *L'output è* 1 *se* $b' = b$*, altrimenti* 0.

Definizione 2.1 (IND-mult-eav)**.** Uno schema a chiave privata $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ ha *cifrature multiple indistinguibili* in presenza di un eavesdropper se per ogni PPT $\mathcal{A}$ esiste una funzione trascurabile negl tale che

$$\Pr\left[\mathsf{PrivK}^{\text{eav-mult}}_{\mathcal{A},\Pi}(n) = 1\right] \le \tfrac{1}{2} + \mathsf{negl}(n),$$

con probabilità calcolata su: casualità di $\mathcal{A}$, dello sperimento, scelta della chiave, del bit b, e i bit casuali di $\mathsf{Enc}_k(\cdot)$.

2.2 Motivazione e relazione con il caso singolo

Contenuto slide. *Osservazione: IND-mult-eav implica IND-eav (caso speciale* $t = 1$*). Non vale l'inverso: controesempio con OTP.*

Esempio 2.2 (OTP non è IND-mult-eav). ***Contenuto slide.*** *Scegli* $M_0 = (0^\ell, 0^\ell)$*,* $M_1 = (0^\ell, 1^\ell)$*. Ricevuto* $c = (c_1, c_2)$*, se* $c_1 = c_2$ *emetti* $b' = 0$*, altrimenti* $b' = 1$*. Poiché OTP è deterministico dato* k*, si ha successo con probabilità* 1*; dunque non è IND-mult-eav.*

Teorema 2.3 (Determinismo $\Rightarrow$ non IND-mult-eav)**.** *Se* Enc *è deterministica, lo schema non può essere IND-mult-eav sicuro.*

Spiegazione del docente. Idea chiave: la ripetizione dello stesso messaggio produce lo stesso cifrato. Per ottenere IND-mult occorre cifratura probabilistica (nonce/IV casuale o simile), pur mantenendo correttezza in decrittazione: si include nel cifrato l'IV/nonce.

3 Attacchi chosen-plaintext (CPA) e modellazione con oracoli

3.1 Oracolo di cifratura e esperimento IND-CPA

Contenuto slide. *Oracolo* $O(\cdot)$*: scatola nera che su query* m *risponde con* $\mathsf{Enc}_k(m)$ *per una chiave segreta* k*. Se* Enc *è randomizzata, usa nuovi bit casuali per ogni query. L'avversario può fare query adattive.*

Contenuto slide. *Esperimento* $\mathsf{PrivK}^{cpa}_{\mathcal{A},\Pi}(n)$*:*

1. Chall *genera* $k \leftarrow \mathsf{Gen}(1^n)$ *e istanzia l'oracolo* $O(m) = \mathsf{Enc}_k(m)$*.*
2. $\mathcal{A}^{O(\cdot)}(1^n)$ *emette* m_0, m_1 *con* $|m_0| = |m_1|$*.*
3. Chall *sceglie* $b \leftarrow \{0,1\}$ *e* $c \leftarrow \mathsf{Enc}_k(m_b)$*.*
4. $\mathcal{A}^{O(\cdot)}$ *riceve* c *(può continuare a interrogare* O*) e restituisce* b'*.*
5. *Vince se* $b' = b$*.*

Definizione 3.1 (IND-CPA (messaggio singolo))**.** Π è CPA-sicuro se per ogni PPT $\mathcal{A}$ esiste negl tale che

$$\Pr\left[\mathsf{PrivK}^{\text{cpa}}_{\mathcal{A},\Pi}(n) = 1\right] \leq \tfrac{1}{2} + \mathsf{negl}(n).$$

3.2 IND-CPA per messaggi multipli con oracolo Left-or-Right

Contenuto slide. *Oracolo* $\mathsf{LR}_{k,b}(m_0, m_1) = \mathsf{Enc}_k(m_b)$*;* b *è fissato all'inizio. Esperimento* $\mathsf{PrivK}^{LR\text{-}cpa}_{\mathcal{A},\Pi}(n)$*:*

1. Chall *genera* $k \leftarrow \mathsf{Gen}(1^n)$ *e* $b \leftarrow \{0,1\}$*.*
2. $\mathcal{A}^{\mathsf{LR}_{k,b}(\cdot,\cdot)}(1^n)$ *interagisce adattivamente e poi emette* b'*.*
3. *Vince se* $b' = b$*.*

Differenze: le coppie $(m_{0,i}, m_{1,i})$ *sono scelte adattivamente; con query* (m, m) *si ottengono cifrature di messaggi scelti (modellando la fase "raccolta coppie" dell'attacco CPA).*

Definizione 3.2 (IND-CPA per cifrature multiple)**.** Π è CPA-sicuro per cifrature multiple se per ogni PPT $\mathcal{A}$ esiste negl tale che

$$\Pr\left[\mathsf{PrivK}^{\text{LR-cpa}}_{\mathcal{A},\Pi}(n) = 1\right] \leq \tfrac{1}{2} + \mathsf{negl}(n).$$

Teorema 3.3 (Equivalenza singolo/multiplo per CPA)**.** *Ogni schema a chiave privata è CPA-sicuro per messaggi multipli se e solo se è CPA-sicuro per messaggi singoli.*

Contenuto slide. *Conseguenza: basta provare IND-CPA nel caso singolo; si ottiene gratis la sicurezza per cifrature multiple. Inoltre, se* Π *è IND-CPA per messaggi di 1 bit, si può ottenere uno schema* Π' *IND-CPA per messaggi di lunghezza arbitraria concatenando cifrature per-bit.*

Spiegazione del docente. Nota pratica: la costruzione per-bit è teorica e inefficiente; nella pratica si usano modalità di operazione di PRP/PRF o schemi AEAD. L'oracolo LR dà più potere all'avversario rispetto al modello "liste statiche", ma la definizione resta equivalente.

3.3 Esempi e motivazioni reali

Contenuto slide. *Esempi storici di known/chosen-plaintext: Seconda Guerra Mondiale (mine posizionate e comunicate; Midway 1942). Esempio moderno: terminali che cifrano input utente prima dell'invio; un avversario può interagire con il terminale e ottenere coppie (messaggio, cifrato).*

4 Funzioni pseudocasuali (PRF)

4.1 Definizioni

Contenuto slide. *Funzione con chiave efficiente:* $F : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$*, con* $F_k(x) = F(k, x)$*. Parametrizziamo lunghezze con il parametro di sicurezza* n*:* $\ell_{key}(n)$*,* $\ell_{in}(n)$*,* $\ell_{out}(n)$*. Caso classico:* $\ell_{key} = \ell_{in} = \ell_{out} = n$*.*

Definizione 4.1 (PRF)**.** Sia F che preserva la lunghezza. F è pseudocasuale se, per ogni PPT distinguisher D,

$$\left|\Pr\left[D^{F_k(\cdot)}(1^n) = 1\right] - \Pr\left[D^{f(\cdot)}(1^n) = 1\right]\right| \leq \mathsf{negl}(n),$$

dove $k \leftarrow \{0,1\}^n$ è uniforme, e f è uniforme in $\mathrm{Func}_n = \{f : \{0,1\}^n \to \{0,1\}^n\}$.

Osservazione 4.2. Il distinguisher non conosce k. Conoscerla renderebbe banale distinguere chiedendo valutazioni e confrontando con F_k noto.

4.2 Scelta casuale di una funzione

> ***Contenuto slide.*** $|\mathrm{Func}_n| = 2^{n \cdot 2^n}$. *Una funzione casuale può essere vista come una tabella di* 2^n *righe riempita on-demand quando si vedono nuovi input.*

4.3 Esempio di non-PRF

Esempio 4.3. $F(k,x) = k \oplus x$ non è PRF: chiedendo x_1, x_2 e ottenendo y_1, y_2, se $y_1 \oplus y_2 = x_1 \oplus x_2$ output 1; ciò accade sempre contro F_k, e solo con prob. 2^{-n} contro f uniforme.

5 Permutazioni pseudocasuali (PRP) e PRP forti

5.1 Definizione di PRP

> ***Contenuto slide.*** *Sia* Perm_n *l'insieme delle permutazioni su* $\{0,1\}^n$*;* $|\mathrm{Perm}_n| = (2^n)!$. *Una funzione con chiave* F *è permutazione con chiave se per ogni* k *la* F_k *è biunivoca su blocchi di lunghezza* n *ed è efficientemente computabile e invertibile (dato* k*).*

Definizione 5.1 (PRP forte)**.** Una permutazione con chiave F che preserva la lunghezza è *pseudocasuale forte* se, per ogni PPT D,

$$\left| \Pr\left[D^{F_k(\cdot),F_k^{-1}(\cdot)}(1^n) = 1\right] - \Pr\left[D^{f(\cdot),f^{-1}(\cdot)}(1^n) = 1\right] \right| \leq \mathsf{negl}(n),$$

con k uniforme e f uniforme in Perm_n.

Osservazione 5.2. Per blocchi lunghi, una permutazione casuale è indistinguibile da una funzione casuale a meno di collisioni sugli output, che sono trascurabili con numero polinomiale di query. Nella pratica, i cifrari a blocchi mirano a istanziare PRP forti su domini finiti.

6 PRF e PRG: costruzioni e relazioni

6.1 Da PRF a PRG

> ***Contenuto slide.*** *Dato* F *PRF, si costruisce un PRG* $G(s) = F_s(1) \,\|\, F_s(2) \,\|\, \dots \,\|\, F_s(\ell)$ *per ogni* ℓ *desiderato. Se* F_s *fosse rimpiazzato da* f *uniforme, l'output sarebbe uniforme; un distinguisher contro* G *darebbe un distinguisher contro* F.

6.2 Da PRG a PRF (input corto)

> ***Contenuto slide.*** *Sia* $G : \{0,1\}^n \to \{0,1\}^{2^{t(n)} \cdot n}$ *con fattore di espansione* $2^{t(n)}n$. *Per* $t(n) = O(\log n)$ *si può definire* $F_k(i)$ *come la* i*-esima riga di lunghezza* n *dell'output tabellare di* $G(k)$*, per* $i \in \{1, \dots, 2^{t(n)}\}$*: ciò dà una PRF su input di* $t(n)$ *bit.*

Osservazione 6.1. Questa costruzione è efficiente solo se $t(n) = O(\log n)$, così che la lunghezza $2^{t(n)}n$ sia polinomiale in n.

7 Ripasso: schema basato su PRG e prova di sicurezza

> **Spiegazione del docente.** Tecnica di riduzione (ripasso): Se G è un PRG, lo schema di cifratura a flusso $\mathsf{Enc}_k(m) = m \oplus G(k)$ (o con keystream della giusta lunghezza) è IND-eav. Dimostrazione per assurdo: supponiamo esista $\mathcal{A}$ che distingue; allora si costruisce un distinguisher D che riceve una stringa Ω (o pseudocasuale $G(k)$ o uniforme) e simula per $\mathcal{A}$ l'esperimento di indistinguibilità cifrando con Ω. Due casi:

- Ω uniforme: la cifratura è OTP, quindi $\mathcal{A}$ indovina con probabilità 1/2.
- $\Omega = G(k)$: la simulazione è perfetta rispetto allo schema reale, e $\mathcal{A}$ ha vantaggio non trascurabile per ipotesi.

La differenza tra le probabilità di successo nei due casi fornisce un distinguisher non trascurabile contro G, in contraddizione alla pseudocasualità di G.

7.1 Sicurezza concreta

Spiegazione del docente. Se G è (T, ε)-pseudocasuale, la riduzione introduce solo un overhead costante c (lancio moneta, XOR, chiamate a $\mathcal{A}$), quindi otteniamo uno schema $(T - c, \varepsilon)$-sicuro. Esempi: si fissano target concreti (es. $T \approx 2^{80}$, $\varepsilon \leq 2^{-60}$) in base allo stato dell'arte di attacchi e potenza computazionale. La teoria usa funzioni trascurabili; l'analisi concreta fissa limiti numerici per tempi e vantaggi.

8 Note aggiuntive e commenti dal docente

Spiegazione del docente.

- La dimostrazione è tipicamente per assurdo: da un attaccante contro la costruzione si crea un distinguisher/solver contro la primitiva assunta sicura.
- La simulazione deve riprodurre fedelmente l'ambiente per cui l'avversario è progettato (es. scelta casuale del bit di sfida, stessa distribuzione dei cifrati).
- Sulla necessità di cifratura probabilistica per IND-mult: includendo un nonce/IV non riutilizzato nel cifrato si può mantenere correttezza della decrittazione e sicurezza.
- Esempi storici di known/chosen-plaintext rinforzano la rilevanza pratica del modello CPA.

Riferimenti

- Slide: EC-SN-25 "Sicurezza: altre nozioni" (Paolo D'Arco, UNISA, EC-2025).
- Katz, Lindell. Introduction to Modern Cryptography.
- Appunti e trascrizione della lezione 7 (CPA, oracoli, PRF/PRP, sicurezza concreta).