***Chú thích:***
oracle model = mô hình máy băm
message digest = bản tóm tắt thông điệp
trapdoor one-way permutations = phép biến đổi một chiều với hàm trapdoor
signature scheme = hệ thống chữ ký
padding scheme = mô hình đệm

## 8.5. Full Domain Hash

Trong phần 6.9.2, chúng ta đã chỉ ra cách xây dựng hệ thống mã hóa bảo mật khóa công khai từ các phép biến đổi một chiều với hàm trapdoor (trong mô hình máy băm ngẫu nhiên). Các ứng dụng thực tế của các hệ thống này dựa trên Hệ thống mật mã hóa RSA và thay thế máy băm ngẫu nhiên bằng một hàm băm như SHA3-224. Trong phần này, chúng ta sử dụng một phép biến đổi một chiều với hàm trapdoor để xây dựng một hệ thống chữ ký bảo mật trong mô hình máy băm ngẫu nhiên. Hệ thống mà chúng ta trình bày được gọi là Full Domain Hash. Tên của hệ thống này xuất phát từ yêu cầu rằng phạm vi của máy băm ngẫu nhiên phải giống với miền của phép biến đổi một chiều với trapdoor được sử dụng trong hệ thống. Hệ thống này được trình bày như là Cryptosystem 8.6.

---

**Cryptosystem 8.6:** *Full Domain Hash*
Gọi k là số nguyên dương, F là tập các phép biến đổi một chiều trapdoor thỏa mãn $f\colon \{0,1\}^k \to \{0,1\}^k$ với mọi $f \in F$ và $G\colon \{0,1\}^* \to \{0,1\}^k$ là hàm ngẫu nhiên. $P = \{0,1\}^*$ và $A = \{0,1\}^k$ và định nghĩa

$$K = \{(f, f^{-1}, G)\colon f \in F\}$$

Cho khóa $K = (f, f^{-1}, G)$, $f^{-1}$ là private key, $(f, G)$ là public key.

Cho $K = (f, f^{-1}, G)$ và $x \in \{0,1\}^*$ và định nghĩa

$$sig_K(x) = f^{-1}(G(x))$$

Chữ ký $y = (y_1, y_2, \dots, y_k) \in \{0,1\}^k$ trên thông điệp x sẽ được xác nhận như sau:

$$ver_K(x, y) = true \Leftrightarrow f(y) = G(x)$$

---

Full Domain Hash sử dụng phương pháp băm-để-ký (hash-then-sign method) quen thuộc. G(x) là bản tóm tắt thông điệp (message digest) được tạo ra bởi

máy băm ngẫu nhiên G. Hàm $f^{-1}$ được sử dụng để ký tóm tắt thông điệp, và f được sử dụng để xác minh nó.

Hãy tưởng tượng một cài đặt dựa trên RSA cho hệ thống này. Hàm $f^{-1}$ sẽ là hàm ký RSA (tức là, giải mã), và f sẽ là hàm xác minh RSA (tức là, mã hóa). Ví dụ để đảm bảo tính bảo mật, chúng ta cần chọn k = 2048. Giả sử máy băm ngẫu nhiên G được thay thế bằng hàm băm SHA3-224. Hàm băm này tạo ra một bản tóm tắt thông điệp 224 bit, vì vậy phạm vi của hàm băm, tức $\{0,1\}^{224}$, là một tập con rất nhỏ của $\{0,1\}^{k} = \{0,1\}^{2048}$. Trong thực tế, cần phải xác định một mô hình đệm (padding scheme) cụ thể để mở rộng một thông điệp 224 bit thành 2048 bit trước khi áp dụng hàm $f^{-1}$. Điều này thường được thực hiện bằng cách sử dụng một mô hình đệm cố định (xác định trước).

Chúng ta tiếp tục đến phần chứng minh về tính bảo mật, trong đó chúng ta giả định rằng F là một tập hợp các phép biến đổi một chiều với trapdoor và G là một máy băm ngẫu nhiên "miền đầy đủ". (Lưu ý rằng các chứng minh về tính bảo mật chúng ta sẽ trình bày không áp dụng khi máy băm ngẫu nhiên được thay thế bằng một hàm băm được xác định hoàn toàn như SHA3-224.) Có thể chứng minh rằng Full Domain Hash an toàn khi chống lại việc làm giả tồn tại (existential forgery) bằng việc tấn công những message được chọn; tuy nhiên, chúng ta sẽ chỉ chứng minh kết quả dễ dàng hơn là Full Domain Hash an toàn khi chống lại việc làm giả tồn tại (existential forgery) bằng cuộc tấn công chỉ vào khóa (key-only attack).

Như thường lệ, chứng minh tính bảo mật là một loại giảm thiểu (type of reduction). Chúng ta giả định rằng có một kẻ thù (adversary) (ví dụ, một thuật toán ngẫu nhiên, được ký hiệu bởi FDH-FORGE) có khả năng làm giả chữ ký (với một xác suất cụ thể) khi nó được cung cấp khóa công khai và truy cập vào máy băm ngẫu nhiên (nhớ rằng nó có thể truy vấn máy băm ngẫu nhiên để lấy giá trị G(x), nhưng không có thuật toán cụ thể được chỉ định để đánh giá hàm G). FDH-FORGE thực hiện một số lần truy vấn máy băm, ví dụ như $q_h$. Cuối cùng, FDH-FORGE xuất ra một giả mạo hợp lệ với một xác suất cụ thể, được ký hiệu là e.

---

**Algorithm 8.1:** FDH-INVERT($z_0$, $q_h$ )

```
external f
procedure SIMG(x)
        if j > qh
                then return ("failure")
                else if j = j0
                then z ← z0
                else let z ∈ {0, 1} k be chosen at random
         j ← j + 1
        return (z)

main
        choose j0 ∈ {1, . . . , qh} at random
        j ← 1
        insert the code for FDH-FORGE(f) here
        if FDH-FORGE(f) = (x, y)
        then {
                if f(y) = z0
                then return (y)
                else return ("failure")}
```

---

Chúng ta xây dựng một thuật toán, FDH-INVERT, cố gắng đảo ngược các phần tử được chọn ngẫu nhiên $z_0 \in \{0,1\}^k$. Tức là, cho trước $z_0 \in \{0,1\}^k$, chúng ta hy vọng
là FDH-INVERT($z_0$) =$f^{-1}(z_0)$. Bây giờ chúng ta trình bày FDH-INVERT như là Thuật toán 8.1.

Thuật toán 8.1 khá đơn giản. Nó chủ yếu bao gồm việc chạy kẻ thù (running the adversary), FDH-FORGE. Các truy vấn băm được thực hiện bởi FDH-FORGE được xử lý bởi hàm SIMG, một mô phỏng của máy băm ngẫu nhiên. Chúng ta đã giả định rằng FDH-FORGE sẽ thực hiện $q_h$ truy vấn băm, ví dụ như $x_1, \ldots, x_{qh}$. Để cho đơn giản, chúng ta giả định rằng các $x_1$ đều là riêng biệt. (Nếu chúng không phải là riêng biệt, thì chúng ta cần đảm bảo rằng SIMG($x_i$) = SIMG($x_j$) khi $x_i = x_j$. Điều này không khó thực hiện; chỉ đòi hỏi một số công việc ghi chép, như đã thực hiện trong Thuật toán 6.14.) Chúng ta ngẫu nhiên

chọn một truy vấn, ví dụ như truy vấn thứ $j_0$, và định nghĩa SIMG($x_{j_0}$) = $z_0$ ($z_0$ là giá trị mà chúng ta đang đảo ngược). Đối với tất cả các truy vấn khác, giá trị SIMG($x_j$) được chọn là một số ngẫu nhiên. Bởi vì $z_0$ cũng là ngẫu nhiên, nên dễ thấy rằng SIMG không thể phân biệt được so với một máy băm ngẫu nhiên thực sự. Do đó, kết quả là FDH-FORGE xuất ra một thông điệp và một chữ ký giả mạo hợp lệ, được ký hiệu là (x, y), với xác suất e. Sau đó, chúng ta kiểm tra xem f(y) = $z_0$; nếu có, thì y = $f^{-1}(z_0)$ và chúng ta đã thành công trong việc đảo ngược $z_0$.

Nhiệm vụ chính của chúng ta là phân tích xác suất thành công của thuật toán FDH-INVERT dưới dạng một hàm của xác suất thành công e của FDH-FORGE. Chúng ta sẽ giả định rằng e > $2^{-k}$, bởi vì một lựa chọn ngẫu nhiên của y sẽ là một chữ ký hợp lệ cho một thông điệp x với xác suất $2^{-k}$, và chúng ta chỉ quan tâm đến những adversaries có khả năng thành công cao hơn đoán ngẫu nhiên.

Giống như chúng ta đã làm ở trên, chúng ta ký hiệu các truy vấn băm được thực hiện bởi FDH-FORGE bằng $x_1, \ldots, x_{q_h}$, trong đó $x_j$ là truy vấn băm thứ j, $1 \leq j \leq q_h$.

Chúng ta bắt đầu bằng cách viết điều kiện cho xác suất thành công, e, dựa trên việc x có $\in$ {x1, . . . , xqh} hay không:

$$e = \mathbf{Pr}[\text{FDH-FORGE thành công} \wedge (x \in \{x_1, \ldots, x_{q_h}\})] + \mathbf{Pr}[\text{FDH-FORGE thành công} \wedge (x \notin \{x_1, \ldots, x_{q_h}\})]. \quad (8.7)$$

Không khó để thấy rằng:

$$\mathbf{Pr}[\text{FDH-FORGE thành công} \wedge (x \notin \{x_1, \ldots, x_{q_h}\})] = 2^{-k}.$$

Điều này xảy ra vì giá trị (chưa xác định) SIMG(x) có khả năng như nhau để nhận bất kỳ giá trị cụ thể nào trong $\{0, 1\}^k$, và do đó xác suất SIMG(x) = f(y) là

$2^{-k}$. (Đây là nơi chúng ta sử dụng giả thiết rằng hàm băm là một "miền đầy đủ".) Thay thế vào (8.7), chúng ta có:

$$\textbf{Pr}[\text{FDH-FORGE thành công} \wedge (x \in \{x_1, \ldots, x_{q_h}\})] \geq e - 2^{-k}. \quad (8.8)$$

Bây giờ chúng ta xem xét xác suất thành công của FDH-INVERT. Bất đẳng thức tiếp theo là rõ ràng:

$$\textbf{Pr}[\text{FDH-INVERT thành công}] \geq \textbf{Pr}[\text{FDH-FORGE thành công} \wedge (x = x_{j_0})]. \quad (8.9)$$

Quan sát cuối cùng của chúng ta là:

$$\textbf{Pr}[\text{FDH-FORGE thành công} \wedge (x = x_{j_0})] = \frac{1}{x_{q_h}} \times \textbf{Pr}[\text{FDH-FORGE thành công} \wedge (x \in \{x_1, \ldots, x_{q_h}\})]. \quad (8.10)$$

Lưu ý rằng phương trình (8.10) đúng vì có xác suất $\frac{1}{q_h}$ rằng x = $x_{j_0}$, trong trường hợp x $\in \{x_1, \ldots, x_{q_h}\}$. Bây giờ, nếu chúng ta kết hợp (8.8), (8.9), và (8.10), chúng ta sẽ có được giới hạn sau đây:

$$\textbf{Pr}[\text{FDH-INVERT thành công}] \geq \frac{e - 2^{-k}}{q_h}. \quad (8.11)$$

Vì vậy, chúng ta đã có được một giới hạn dưới cụ thể về xác suất thành công của FDH-INVERT. Chúng ta đã chứng minh kết quả sau đây.

**THEOREM 8.1:** Giả sử tồn tại một thuật toán FDH-FORGE có khả năng xuất ra một giả mạo tồn tại cho Full Domain Hash với xác suất e > $2^{-k}$ sau khi thực hiện $q_h$ truy vấn tới máy băm ngẫu nhiên, sử dụng cuộc tấn công chỉ có khóa. Sau đó, tồn tại một thuật toán FDH-INVERT có khả năng tìm ra các phần tử ngẫu nhiên nghịch đảo $z_0 \in \{0, 1\}^k$ với xác suất ít nhất là $\frac{e - 2^{-k}}{q_h}$.