Information theoretically secure post-quantum cryptography

Peter P. Rohde*

CONTENTS

* peter@peterrohde.org; https://www.peterrohde.org

I. LINKS & CHATS

- One way functions and P vs NP: https://en.wikipedia.org/wiki/One-way_function: "The existence of such one-way functions is still an open conjecture. Their existence would prove that the complexity classes P and NP are not equal,"
- GapP, NP, binary optimisation problems: https://chatgpt.com/share/2e9f8c0f-fa1d-4d97-9af0-c195ad3cd22c
- Homomorphism between B_n and S_n: https://chatgpt.com/share/c0390897-77f1-403e-bce9-ca2bf5c9fccc
- Even Hamming weight for $H(x \oplus y) \in 2\mathbb{Z}$ if $H(x) = H(y)$. Hence also for $y = \pi \circ x$ where $\pi \in S_n$.
- Word problem for the braid group B_n is poly-time solvable: https://chatgpt.com/c/f8827503-7af9-4b81-b3dd-a79f1891746e
- Wreath product https://en.wikipedia.org/wiki/Wreath_product
- Braid permutations $\pi : B_n \to S_n$, for $\beta \in B_n$ we have $\pi(\beta)$: https://chatgpt.com/share/6c61f5ef-a33d-4c76-8f1f-0d8bdf1bb4aa

II. OVERVIEW

The goal is to create information-theoretically secure asymmetric PQC primitives where the only hardness assumption is the hardness of brute-force, reducing all arguments to entropic ones.

There are multiple primitives we will combine together.

The purpose of differential encoding is create differential pairs encoded against a secret, where the differential term is public and cannot reveal the secret itself.

Combining differential encoding with random codewords we have a codespace where all codewords are unique (statistical security assumption) and errors on differentials are always detectable (but not correctable).

Permutation codes and their respective algebra are used to create asymmetric primitives from random codewords. This closely relates to and is inspired by hash algebra. Here the central concept is that any permutation on the bits in codewords creates a distinct codeword (statistical security assumption). The asymmetric objects introduced here are defines under the algebra of the symmetric group and their inversion assuming random codewords is only via brute force as the objects are provably non-invertible.

The security of asymmetric objects relates to their preimage space which decomposes into collisional and non-collisional partitions. The former is insecure (ambiguous secrets) while the latter is secure. Quantifying the size of these respective spaces affords provable statements on statistical security alone.

The schemes resulting from this are compositional under binary XOR algebra.

III. DIFFERENTIAL ENCODING

A pair of bit-strings $\{x, y\} \in \{0,1\}^n$ may be expressed differentially using the tuple,

$$\begin{aligned}\llbracket x, y \rrbracket &\equiv [x, x \oplus y] \\ &\equiv [x \oplus y, x]. \end{aligned} \tag{3.1}$$

Here the differential term $x \oplus y$ alone reveals no information about x or y while the non-differential term unlocks the code to reveal both. The validity of differentially encoded tuples may be trivially confirmed given knowledge of both terms.

Differential encodings are composable under $\oplus$,

$$\llbracket x_1, y_1 \rrbracket \oplus \llbracket x_2, y_2 \rrbracket \Rightarrow \llbracket x_1 \oplus x_2, y_1 \oplus y_2 \rrbracket, \tag{3.2}$$

with the properties,

$$\begin{aligned}\llbracket \mathbf{0}, x \oplus y \rrbracket &= \llbracket x, y \rrbracket, \\ \llbracket x, y \rrbracket &\Rightarrow x = y. \end{aligned} \tag{3.3}$$

IV. ENTROPY CODES

Entropy codes employ random codewords,

$$\mathcal{X} : x \in \{0,1\}^n, \tag{4.1}$$

sampled using random variable $\mathcal{X}$ with respective per-bit Shannon entropy,

$$0 \leq H(\mathcal{X}) \leq 1. \tag{4.2}$$

Under maximum entropy $H(\mathcal{X}) = 1$ conditions all codewords are sampled with probability,

$$\mathbb{P}(x) = \frac{1}{2^n}, \tag{4.3}$$

and are orthogonal,

$$x_i \oplus x_j \neq \mathbf{0}. \tag{4.4}$$

Under differential encoding,

$$[x, x \oplus y], \tag{4.5}$$

entropy addition implies the differential term $x \oplus y$ exhibits entropy at least,

$$H(x \oplus y) \geq \min(H(x), H(y)). \tag{4.6}$$

Hence parity terms form unique random codewords.

For known $\{x, y\}$, its differential code $\llbracket x, y \rrbracket$ affords both detection and correction of all bit-errors.

A. Collision space

The collision space of an n-bit function,

$$f_n(x) \to y, \quad x, y \in \{0,1\}^n, \tag{4.7}$$

is the number of images y with multiple pre-images x,

$$\mathrm{Col}(f) = |\{\mathrm{Im}(x) \,|\, \mathrm{Im}(x) = \mathrm{Im}(y \neq x)\}|, \tag{4.8}$$

defining the non-invertible image subspace. In the collision-free subspace all images have single pre-images,

$$\overline{\mathrm{Col}}(f_n) + \mathrm{Col}(f_n) = 2^n, \tag{4.9}$$

paritioning the image space.

Random codes are vulnerable only when multiple secrets map to the same asymmetric encoding,

$$f : x, y \to \langle x \rangle_{\pi}^{\pi^{-1}}, \quad x \neq y. \tag{4.10}$$

Now the collision-free subspace of the public object is defined as secure while the collision space is insecure via pre-image ambiguity.

A birthday attack is a statistical attack on collision space where the goal is to find *any* collision rather than a *specific* collision, affording quadratically enhanced scaling. For an n-bit random bit-string with 2^n possible values, upon taking k random samples the likelihood of finding a collision scales as,

$$\mathbb{P}(n, k) \approx 1 - \exp\left[-\frac{k^2}{2^{n+1}}\right]. \tag{4.11}$$

For $k = \mathrm{poly}(n)$ this ensures the asymptotic statistical security of random codewords x.

The collision space of asymmetric objects of the form $\langle x \rangle_{\pi}^{\pi^{-1}}$ is given by the set of satisfying y such that,

$$\langle x \rangle_{\pi}^{\pi^{-1}} = \langle y \rangle_{\pi}^{\pi^{-1}}, \tag{4.12}$$

the satisfiability problem of finding bit-strings which under a random braid have a given symmetric difference. This presumably mandates a brute-force search of pre-image space given that the group is the symmetric group with no hidden structure.

V. PERMUTATION CODES

We define the algebra,

$$(\mathbb{Z}_2^n \times S_n, \oplus, \pi), \tag{5.1}$$

comprising the bitwise binary $(\mathbb{Z}_2^n, \oplus)$ operator and the unary (S_n, π) operator. Defining permutations over bit-string elements,

$$\begin{aligned} \pi &\in S_n, \\ \pi \circ x &\equiv x_\pi, \end{aligned} \tag{5.2}$$

as unary operators we inherit algebraic properties from the symmetric group.

- *Composition*: $[\pi_i \circ \pi_j] \circ x \equiv x_{\pi_i \circ \pi_j}$.
- *Distributivity*: $\pi \circ (x \oplus y) \equiv [x \oplus y]_\pi \equiv x_\pi \oplus y_\pi$.

For unknown x, the permuted bit-string x_π can be decoded to x given π.

Structures of the form,

$$x_\pi x \equiv x_\pi \oplus x, \tag{5.3}$$

cannot be decoded from π alone. However, the action of x or x_π on $x_\pi x$ reveals the other,

$$\begin{aligned} x \oplus (x_\pi \oplus x) &= x_\pi, \\ x_\pi \oplus (x_\pi \oplus x) &= x. \end{aligned} \tag{5.4}$$

Via distributivity we have the additional symmetric property,

$$\pi^{-1} \circ (x_\pi x) = x_{\pi^{-1}} x. \tag{5.5}$$

Differential codes preserve their differential structure under common global permutations but not under non-uniform permutations,

$$\begin{aligned} \pi \circ [\![x, y]\!] &\equiv [\![x_\pi, y_\pi]\!], \\ [\![x_\pi, y_\pi]\!] &\sim [\![x, y]\!], \\ [\![x, y_\pi]\!] &\not\sim [\![x, y]\!] \\ [\![x_\pi, y]\!] &\not\sim [\![x, y]\!] \end{aligned} \tag{5.6}$$

VI. ASYMMETRIC CODES

Consider objects $s_\pi s$ and $m_\pi \cdot h(m)$, both public, where Alice knows secret s and Alice knows secret m.

$$[s_\pi s] \cdot [m_\pi h(m)] = [s \cdot m] \cdot [s_\pi \cdot h(m_\pi)] \tag{6.1}$$

We use the notation,

$$\begin{aligned} x_{\pi_1} y_{\pi_2} &\equiv \langle x, y \rangle_{\pi_1}^{\pi_2}, \\ x_\pi x_{\pi^{-1}} &\equiv \langle x \rangle_\pi^{\pi^{-1}}. \end{aligned} \tag{6.2}$$

If a π index is not written it's implied the identity permutation π_0.

This encoding under pairs of permutations implies a braid representation.

Sloppily we'll also use $x \oplus y \equiv x \cdot y \equiv xy$ as shorthand.

A. One-way functions

For secret x we define the public object,

$$x \xrightarrow{\pi} \langle x \rangle_\pi^{\pi^{-1}}. \tag{6.3}$$

This defines a one-way function enabling verification given x and π but does not allow reverse evaluation of x even if π is known via pre-image resistance.

B. Pre-image resistance

Let,

$$\mathcal{X} : x \in \{0,1\}^n, \quad \text{s.t. } w(x) = n/2, \tag{6.4}$$

be a random bit-string with Hamming weight $w(x) = n/2$. The automorphisms of x are equivalent to all permutations over the subsets of elements with the same bit-value,

$$\begin{aligned} \mathrm{Aut}(x) &= \mathrm{Sym}(x^{(0)}) \times \mathrm{Sym}(x^{(1)}) \\ &= S_{n/2} \times S_{n/2}. \end{aligned} \tag{6.5}$$

where,

$$x^{(k)} = \{i \,|\, x_i = k\}_{i \in x}. \tag{6.6}$$

The space of x where $w(x) = n/2$ has order,

$$\frac{|\mathrm{Sym}(x)|}{|\mathrm{Aut}(x)|} = \frac{n!}{(\frac{n}{2})!(\frac{n}{2})!} = \binom{n}{n/2}. \tag{6.7}$$

All x where $w(x) = n/2$ are related by permutations. Hence, for given x and random $\pi \in S_n$, x_π is an independent random bit-string with $w(x_\pi) = w(x) = n/2$.

Let,

$$x_\pi x = c, \tag{6.8}$$

be an asymmetric permutational primitive where c is known and we want to find a satisfying pre-image x.

Consider the table,

$$[x_\pi, \tilde{x}, \tilde{x} \oplus \tilde{c}]_i, \tag{6.9}$$

with rows over i, row-ordered such that the first block of $n/2$ rows have $x_i = 0$ and the second block have $x_i = 1$ (tilde denotes this ordering). The second and third columns are mutually order and under multiplication provide $\tilde{c}$. This is preserved under any permutation within blocks but not between blocks.

The pre-image solution corresponds to choosing x_π such that multiplying the first column by the third column yields $\tilde{c}$.

C. One-way functions

The one-way function from Eq. (6.3) acts distributively over $\oplus$,

$$x \oplus y \xrightarrow{\pi} \langle x \rangle_{\pi}^{\pi^{-1}} \oplus \langle y \rangle_{\pi}^{\pi^{-1}}. \tag{6.10}$$

For known x and unknown y the object $y_\pi y$ acts as a one-time zero-knowledge proof of y. The proof is one-time as it is the same for all x.

D. Hash function model

Using the asymmetric model,

$$h : x \oplus \langle y \rangle_{\pi}^{\pi^{-1}} \xrightarrow{\pi} \langle x \rangle_{\pi}^{\pi^{-1}} \oplus y, \tag{6.11}$$

the $\langle y, y \rangle_{\pi}^{\pi^{-1}}$ term acts a known pre-image modulation based on the hidden underlying action of unknown $\{y, \pi\}$, reproducing the property of hash pre-image salting. This provides a model for a deterministic pseudo-random hash function with hidden evaluation, while remaining verifiable.

This model implies the hash function cannot be collision free even over the same input and output domains. While in principle a function can be bijective over matching input and output domains the hash function model cannot be.

There may be an inherent underlying connection here via complexity arguments. Uniquely encoding arbitrary bijective maps $\{0,1\}^n \to \{0,1\}^n$ has $O(2^n)$ space and time complexity. With polynomial time/space encoding this is unachievable if the function is modelled as random and unstructured.

* Collision space vanishes with $n \to \infty$, hence hash function model becomes bijective in the infinite limit.

E. TCF model

Using the asymmetric model,

$$f : x \oplus \langle x_0 x_1 \rangle_{\pi}^{\pi^{-1}} \xrightarrow{\pi} \langle x \rangle_{\pi}^{\pi^{-1}} \oplus \langle x_0, x_1 \rangle_{\pi}^{\pi^{-1}}. \tag{6.12}$$

$$\begin{aligned} f(x_0) &= x_0^{\pi} x_0^{\pi^{-1}} \cdot x_0^{\pi} x_1^{\pi^{-1}} = x_0^{\pi^{-1}} x_1^{\pi^{-1}}, \\ f(x_1) &= x_1^{\pi} x_1^{\pi^{-1}} \cdot x_0^{\pi} x_1^{\pi^{-1}} = x_0^{\pi} x_1^{\pi}. \end{aligned} \tag{6.13}$$

$$\begin{aligned} f(x) &= x^{\pi} x \cdot x_0 x_1, \\ f(x_0) &= x_0^{\pi} x_0^{\pi^{-1}} \cdot x_0^{\pi} x_1^{\pi^{-1}} = x_0^{\pi^{-1}} x_1^{\pi^{-1}}, \\ f(x_1) &= x_1^{\pi} x_1^{\pi^{-1}} \cdot x_0^{\pi} x_1^{\pi^{-1}} = x_0^{\pi} x_1^{\pi}. \end{aligned} \tag{6.14}$$

VII. ENCRYTPION

Given secrets $\{x, m\}$ with respective public objects $\{\langle x \rangle_{\pi^{-1}}^{\pi}, \langle m \rangle_{\pi^{-1}}^{\pi}\}$, public information can be expressed,

$$\begin{aligned} \langle x \rangle_{\pi^{-1}}^{\pi} \cdot \langle m \rangle_{\pi^{-1}}^{\pi} &= \langle m \cdot x \rangle_{\pi^{-1}}^{\pi} \\ &= x_\pi x_{\pi^{-1}} \cdot m_\pi m_{\pi^{-1}}. \end{aligned} \tag{7.1}$$

Now the solution revealing m_π is,

$$\texttt{sol} = x_\pi x_{\pi^{-1}} m_{\pi^{-1}}, \tag{7.2}$$

hence unlocking m under known π.

Applying known π yields public,

$$x_{\pi^2} x \cdot m_{\pi^2} m. \tag{7.3}$$

Provision of,

$$\texttt{sig} = m_{\pi^2} \cdot x_{\pi^2} \cdot x, \tag{7.4}$$

affords,

$$\texttt{sig} \oplus \pi \circ \langle x \oplus m \rangle_{\pi^{-1}}^{\pi} = m. \tag{7.5}$$

$\langle x, y \rangle_{\pi}^{\pi^{-1}}$

An asymmetric trapdoor function on the differential $x \oplus y$ defined as,

$$[x, x \oplus y] \xrightarrow{\pi} [x, x \oplus y_\pi]. \tag{7.6}$$

For known π, knowing x

π-symmeterised objects are pre-image resistant and public-safe, while their pre-images are not.

Via composition, differential codes for π-symmeterised objects preserve differentiality of their pre-image codes,

$$[\![x, y]\!] \Rightarrow [\![x_\pi x, y_\pi y]\!]. \tag{7.7}$$

Defining a secret key s and public key $s \oplus p$ differentially we obtain,

$$\begin{aligned} [\![s, p]\!] &= [s, s \oplus p], \\ [s, s \oplus p] &= [p \oplus s, p], \\ [\![s_\pi s, p_\pi p]\!] &= [s \oplus p, s_\pi \oplus p_\pi], \end{aligned} \tag{7.8}$$

Similarly defining a message m differentially encoded against the same secret key s we have,

$$[\![s, m]\!] = [s, s \oplus m]. \tag{7.9}$$

Under composition this implies,

$$[s, s \oplus p] \oplus [s, s \oplus m] = [\mathbf{0}, p \oplus m]. \tag{7.10}$$

Hence treating differential terms as shared information affords evaluation of $p \oplus m$ independent of s.

Introduce the term asymmetric term $m_\pi h(m)$ encoding a permuted message against its hash.

Let us introduce symmetrised public terms,

$$\begin{aligned} &s_\pi s, \\ &p_\pi p. \end{aligned} \tag{7.11}$$

Using the permuted differential codes we have the identity,

$$[s, s{\oplus}p]{\oplus}[s, s_\pi{\oplus}m_\pi] = [\mathbf{0}, s_\pi s{\oplus}m_\pi p]. \tag{7.12}$$

$$[s, s{\oplus}p]{\oplus}[p, p_\pi{\oplus}m_\pi] = [\mathtt{pk}, p_\pi s{\oplus}m_\pi p]. \tag{7.13}$$

If the differential term is publicly shared this affords decoding under,

VIII. OLD

We define asymmetric key-pairs via their differential encoding,

$$\mathtt{sk}, \mathtt{pk} \in \{0,1\}^n, \tag{8.1}$$

$$\begin{aligned} \mathtt{kp} &= [\![\mathtt{sk}, \mathtt{pk}]\!] \\ &= [\mathtt{sk}, \mathtt{sk}{\oplus}\mathtt{pk}], \end{aligned} \tag{8.2}$$

IX. DIGITAL SIGNATURES

Generate signature,

$$\begin{aligned} \mathtt{sig}(m, \mathtt{sk}) &= [\![\mathtt{sk}, \mathtt{pk}]\!]{\oplus}[\![\mathtt{sk}, m]\!] \\ &= [\![\mathbf{0}, \mathtt{pk}{\oplus}m]\!] \\ &= \mathtt{pk}{\oplus}m. \end{aligned} \tag{9.1}$$

Verify signature,

$$\begin{aligned} \mathtt{ver}(\mathtt{sig}, \mathtt{pk}) &= \Delta(m, \mathtt{pk}){\oplus}\Delta() \\ &= \mathtt{pk}. \end{aligned} \tag{9.2}$$

X. ASYMMETRIC ENCRYPTION

Encode,

$$\begin{aligned} \tilde{m} &= \mathtt{enc}(m, \mathtt{pk}) \\ &= [\![m, \mathtt{sk}]\!] \\ &= \mathtt{sk}{\oplus}m. \end{aligned} \tag{10.1}$$

Decode,

$$\begin{aligned} \mathtt{dec}(\tilde{m}, \mathtt{sk}) &= \tilde{m}{\oplus}\bar{\mathtt{kp}} \\ &= \mathtt{enc}(\mathtt{sk}, m){\oplus}[\![\mathtt{pk}, \mathtt{sk}]\!] \\ &= \mathtt{sk}{\oplus}m. \end{aligned} \tag{10.2}$$

XI. AUTHENTICATION

* $auth = (s1 + p1) + (s2 + p2) = p1 + p2$

XII. KEY REUSE

Consider multiple messages m_i signed by the same public key,

$$\mathtt{enc}(m_i, \mathtt{pk}) = \mathtt{pk}{\oplus}m_i, \tag{12.1}$$

Verify signature,

$$\begin{aligned} \mathtt{ver}(\mathtt{sig}, \mathtt{pk}) &= \Delta(m, \mathtt{pk}){\oplus}\Delta() \\ &= \mathtt{pk}. \end{aligned} \tag{12.2}$$

A. Key-establishment & secret sharing

XIII. OLD STUFF

XIV. DIFFERENTIAL HASH CODES

A pair of bit-strings $\{x, y\}$ may be expressed differentially using the tuple,

$$[x, x{\oplus}y]_{\oplus}, \tag{14.1}$$

where the differential term $x{\oplus}y$ alone reveals no information about x or y while the non-differential term unlocks the code to reveal both. The validity of differentially encoded tuples may be trivially confirmed given knowledge of both terms.

We define the differential hash operators,

$$\begin{aligned} \Delta(x) &= h(x){\oplus}x, \\ \Delta_\pi(x) &= h(x_\pi){\oplus}x, \end{aligned} \tag{14.2}$$

where $\pi \in S_n$ for $x \in \{0,1\}^n$ is a permutation over the elements of x. These encode a hash's image and pre-image together while revealing neither assuming hash pre-image resistance. We have the properties,

$$\begin{aligned} h(x) &= \Delta(x){\oplus}x, \\ x &= \Delta(x){\oplus}h(x). \end{aligned} \tag{14.3}$$

The Δ operator inherits pre-image resistance from $h(\cdot)$. Knowing $\Delta(x)$ alone reveals neither x nor $h(x)$, however additionally knowing x or $h(x)$ enables verification of $\Delta(x)$. Finding x for given $\Delta(x)$ reduces to the pre-image resistance of the hash function $h(\cdot)$.

The non-differentially encoded tuple $\{x, h(x)\}$ allows x to unlock $h(x)$, while $h(x)$ cannot unlock x. The second element reveals $h(x)$ alone, but not x via pre-image resistance. Under the differential encoding,

$$[x, \Delta(x)]_{\oplus} = [x, h(x){\oplus}x]_{\oplus}, \tag{14.4}$$

the second element reveals neither x nor $h(x)$, while the first element reveals both, given that $h(x)$ can be efficiently forward-evaluated. Alternately, under the differential encoding,

$$[h(x), \Delta(x)] = [h(x), h(x) \oplus x], \tag{14.5}$$

the non-differential term $h(x)$ affords unlocking the code but does not on its own reveal x via hash pre-image resistance. Under both encodings knowing either x or $h(x)$ alone enables verification.

The differential operator is distributive only over its unhashed components,

$$\begin{aligned} \Delta(x \oplus y) &= h(x \oplus y) \oplus x \oplus y \\ \Delta(x) \oplus \Delta(y) &= h(x) \oplus h(y) \oplus x \oplus y. \end{aligned} \tag{14.6}$$

The symmetric difference between $\Delta(x \oplus y)$ and $\Delta(x) \oplus \Delta(y)$ gives the 'distributor' (equivalent of commutator for distributivity),

$$\Delta(x \oplus y) \oplus \Delta(x) \oplus \Delta(y) = h(x \oplus y) \oplus h(x) \oplus h(y), \tag{14.7}$$

defining the distributivity of Δ operator over the action of $\oplus$.

Standard differential codes are composable,

$$\begin{aligned} &[x, x \oplus y)]_{\oplus} \oplus [x', x' \oplus y']_{\oplus} \\ &\sim [x \oplus x', x \oplus y \oplus x' \oplus y']_{\oplus}. \end{aligned} \tag{14.8}$$

For differential hash codes,

$$\begin{aligned} &[x, \Delta(x)]_{\oplus}, \\ &[y, \Delta(y)]_{\oplus}, \end{aligned} \tag{14.9}$$

we have distinct composition rules,

$$\begin{aligned} &[x \oplus y, \Delta(x \oplus y)]_H, \\ &[x \oplus y, \Delta(x) \oplus \Delta(y))]_{\oplus}. \end{aligned} \tag{14.10}$$

The $[\cdot, \cdot]_H$ composition is verifiable by hashing the left hand term. The $[\cdot, \cdot]_{\oplus}$ composition is not hash-verifiable but preserves all differential encoding constraints.

Permutations π are distributive over $\oplus$ but not commutative,

$$\begin{aligned} &\pi(x \oplus y) = \pi(x) \oplus \pi(y), \\ &\pi(x) \oplus y \neq x \oplus \pi(y), \end{aligned} \tag{14.11}$$

whereas $\oplus$ is commutative but not distributive (in general, depending on parity of number of terms under distribution),

$$x \oplus y = y \oplus x. \tag{14.12}$$

- $h(m \oplus s)$ will reveal the private $h(s)$ for chosen $m = \mathbf{0}$.
- $h(m \oplus s \oplus x)$ will reveal the private $h(x)$ for chosen $m = x$ if x is public.
- $h(\pi(m \oplus s)) = h(m_\pi \oplus s_\pi)$ can only reveal $h(s_\pi)$ (not secret) for public π and chosen m, but cannot reveal secret $h(s)$.

XV. ASYMMETRIC CODES

We define key-pairs as,

$$\begin{aligned} \mathtt{sk} &\in \{0,1\}^n, \\ \mathtt{pk} &= \{\mathtt{pk}_\Delta, \mathtt{pk}_\pi\}, \\ \mathtt{pk}_\Delta &= \Delta(\mathtt{sk}), \\ \mathtt{pk}_\pi &\in S_n, \end{aligned} \tag{15.1}$$

where $\mathtt{sk}$ be a secret bit-string, $h(\{0,1\}^n) \to \{0,1\}^n$ an n-bit endomorphic hash function, and $\pi \in S_n$ a permutation on n bits. Since $|S_n| = n!$ encoding π requires $\lceil \log_2(n!) \rceil$ bits. For $n = 256$ we have $\lceil \log_2(n!) \rceil = 1684$ bits.

Since $\mathtt{pk} = \Delta(\mathtt{sk})$ is public both $\mathtt{sk}$ and $h(\mathtt{sk})$ must be private to prevent unlocking the public key, both acting as trapdoors for the differential encoding.

$$[m_\pi \oplus s_\pi, \Delta_\pi(m_\pi \oplus s_\pi)], \tag{15.2}$$

$$\begin{aligned} \Delta_\pi(m_\pi \oplus s_\pi) &= \Delta_\pi(m_\pi)\Delta_\pi(s_\pi) \\ &\oplus h(m_\pi) \oplus h(s_\pi) \oplus h(m_\pi \oplus s_\pi) \end{aligned} \tag{15.3}$$

XVI. DIGITAL SIGNATURES

To sign message $m \in \{0,1\}^n$ Alice makes public the differentially encoded, signed message $\Delta(m_\pi)$ and signature,

$$\mathtt{sig}_\pi(m) = h(m_\pi \oplus \mathtt{sk}_\pi). \tag{16.1}$$

The signature has the property that when combined with public information it reveals the hash of the message being signed,

$$\mathtt{sig}_\pi(m) \oplus \Delta(m) \oplus \Delta(\mathtt{sk}_\pi) = h(m). \tag{16.2}$$

Employing the modulated public key $\Delta(\mathtt{sk}_\pi)$,

$$\mathtt{sk}_\pi \equiv \mathtt{sk} \oplus h(\mathtt{pk}), \tag{16.3}$$

prevents the signature from revealing the trapdoor $h(\mathtt{sk})$ which unlocks the public key,

$$\begin{aligned} h(\mathtt{sk}) \oplus \Delta(\mathtt{sk}) &= \mathtt{sk}, \\ h(\mathtt{sk}_\pi) \oplus \Delta(\mathtt{sk}) &\neq \mathtt{sk}. \end{aligned} \tag{16.4}$$

XVII. ENCRYPTION

For asymmetric encryption we reverse the roles of m and $h(m)$. Bob wishes to send message m to Alice and makes public,

$$\mathtt{enc}(m) = \{\Delta(m), h(m)\}. \tag{17.1}$$

Alice now decrypts using the message hash $h(m)$ to reveal the original message,

$$\Delta(s_\pi) \oplus \Delta(m) \oplus h(m) = m. \tag{17.2}$$

A. Multi-sigs

Signatures are commutative, additive and composable.

$$\texttt{sig}_\pi(m) = \Delta(s_\pi) \oplus h(m), \tag{17.3}$$

B. Key-establishment & secret sharing

Using the asymmetric encryption protocol, Alice finally communicates the verification hash,

$$\texttt{key} = h(\texttt{sk} \oplus m), \tag{17.4}$$

back to Bob, who also able to verify its validity. The verification hash now provides confirmation of a jointly prepared hash-based random number given by the XOR-salted hash of Alice's secret key and Bob's chosen salt, which cannot be spoofed by either.

XVIII. NOTES

* The hash collision space translates to decoding failure.

Differentially encoded tuples are composable under bitwise XOR,

$$[x, \Delta(x)] \oplus [y, \Delta(y)] = [x \oplus y, \Delta(x) \oplus \Delta(y)], \tag{18.1}$$

via the commutativity of $\oplus$.

Compare above rhs,

$$h(x \oplus y) \oplus x \oplus y = h(x \oplus y) \oplus h(x) \oplus h(y). \tag{18.2}$$

π known by inverting π and multiplying the tuple elements to confirm consistency. For unknown or incorrect π hash verification fails.

The bit permutation operator, $\pi(\cdot)$, is distribute over the bit-wise XOR operator, $\oplus$,

$$\pi(x) \oplus \pi(y) = \pi(x \oplus y). \tag{18.3}$$

Hence differential encoding relationships are preserved under the uniform action of π,

$$[x, x \oplus y] \sim [\pi(x), \pi(x \oplus y)], \tag{18.4}$$

but are in general not preserved under non-uniform action of π,

$$[x, x \oplus y] \not\sim [\pi(x), x \oplus y]. \tag{18.5}$$

We'll employ the shorthand,

$$\pi \circ [x, y] = [\pi(x), \pi(y)]., \tag{18.6}$$

to denote the uniform action of π over a differential code.

While permutations are distributive over $\oplus$ they do not commute through hashes,

$$\pi(h(x)) \neq h(\pi(x)). \tag{18.7}$$

$$\Delta(\pi(x \oplus y)) = h(\pi(x \oplus y)) \oplus \pi(x) \oplus \pi(y). \tag{18.8}$$

A. Symmetric encryption

Consider a single shared secret random bit-string differentially encoded against unique, single-use random bit-strings, also secret,

$$\begin{aligned} s &\in \{0,1\}^n,, \\ \mathcal{X}_i &\in \{0,1\}^n, \\ &[s, s \oplus \mathcal{X}_i]. \end{aligned} \tag{18.9}$$

Two parties $\{i, j\}$ publicly share their differential terms yielding the respective private and public terms,

$$\begin{aligned} \texttt{Priv} &: \{s, \mathcal{X}_i, \mathcal{X}_j\}, \\ \texttt{Pub} &: \{s \oplus \mathcal{X}_i, s \oplus \mathcal{X}_j, \mathcal{X}_i \oplus \mathcal{X}_j\}, \end{aligned} \tag{18.10}$$

where,

$$\mathcal{X}_{i,j} = \mathcal{X}_i \oplus \mathcal{X}_j, \tag{18.11}$$

is a jointly established publicly-known random variable obtained under addition of both parties' communicated differential terms.

If i wishes to send a message to j they prepare,

$$[s, s \oplus m \oplus \mathcal{X}_i], \tag{18.12}$$

which can be differentially decoded from s given $\mathcal{X}_i$.

Publicly communicating the differential term, the publicly and privately known objects are,

$$\begin{aligned} \texttt{Priv} &: \{s, m, \mathcal{X}_i, \mathcal{X}_j\}, \\ \texttt{Pub} &: \{s \oplus \mathcal{X}_i, s \oplus \mathcal{X}_j, \mathcal{X}_{i,j}, \\ &\quad s \oplus m \oplus \mathcal{X}_i, s \oplus m \oplus \mathcal{X}_j, \\ &\quad m \oplus \mathcal{X}_j, m \oplus \mathcal{X}_i\}. \end{aligned} \tag{18.13}$$

Now public information is unable to reveal m or s while the additional private information can.

The spaces of private and public information form groups with group generators given by privately held and publicly shared objects,

$$\begin{aligned} G_{\text{priv}} &\sim \text{GF}(11) \sim \langle s, m, \mathcal{X}_i, \mathcal{X}_j \rangle, \\ G_{\text{pub}} &\sim \text{GF}(7) \sim \langle s \oplus \mathcal{X}_i, s \oplus \mathcal{X}_j, m \oplus \mathcal{X}_i \rangle. \end{aligned} \tag{18.14}$$

Upon publicly communicating only the differential terms other parties in possession of s can decode all x_i. Upon repeating using independent samples of x_i but with the same secret Eve is in possession of $s \oplus x_i$ for each sample. Combining multiple interceptions Eve can produce,

$$s \oplus x_i \oplus s \oplus x_j = x_i \oplus x_j, \tag{18.15}$$

which is independent of s and reveals no information about s in the absence of knowing any x_i. Thus over n iterations the parties securely share $\{x\}$ where all x_i are statistically independent, maximum entropy random

variables. Concatenation enables a finite length shared secret to facilitate the secure sharing of arbitrarily long random bit-strings which can subsequently be employed in a one-time pad cipher, affording statistical security bounded by the entropy rate.

The security of $s \oplus x$ can be information-theoretically interpreted in terms of the mutual information between the secret s and public information in $s \oplus x$. For random variables X and Y their mutual information is given by,

$$I(X;Y) = H(X) + H(Y) - H(X,Y). \tag{18.16}$$

Let,

$$\begin{aligned} X &\equiv \{s, x\}, \\ Y &\equiv s \oplus x, \end{aligned} \tag{18.17}$$

where X denotes private information and Y public information. We have the identities,

$$\begin{aligned} H(X) &\leq H(Y) \leq 1, \\ H(X,Y) &= 2H(X), \\ I(X;Y) &= 2H(X) - H(X,Y) \\ &= 0, \\ I(E) &= I(X_s;Y) = I(X_x;Y) \\ &= H(X) + H(Y) - H(X,Y) \\ &= H(Y) - H(X) \\ &\leq 1 - H(X). \end{aligned} \tag{18.18}$$

Hence the mutual information between either private random variables and the encoded public random variable, equivalently the extractable information by an eavesdropper, is upper bounded by $1 - H(X)$ where $0 \leq H(X) \leq 1$. For perfect entropy sources where $H(X) = 1$ this implies zero information leakage to Eve.

1. Quantum key distribution

QKD can be equivalently conceptualised. Consider the entanglement-based E91 QKD protocol where for each Bell pair Alice and Bob choose random measurement bases with random measurement outcomes,

$$\begin{aligned} b_{A,B} &\in \{Z \equiv 0, X \equiv 1\}, \\ m_{A,B} &\in \{0, 1\}. \end{aligned} \tag{18.19}$$

Post-selecting on outcomes where the measurement bases chosen by Alice and Bob are consistent, their respective measurement outcomes are dictated by parity constraints,

$$m_A \oplus m_B = p_b, \quad (b_A = b_B,\, p_b \in \{0, 1\}). \tag{18.20}$$

where p_b is imposed by the choice of Bell pair and measurement basis. Under this model for QKD security is statistical and associated with the entropy of b and m, mediated by entanglement enforcing their parity constraints.

XIX. DIFFERENTIAL HASH CODES

A pair of bit-strings $\{x, y\}$ may be expressed differentially using the tuple,

$$[x, x \oplus y]_{\oplus}, \tag{19.1}$$

where the differential term $x \oplus y$ alone reveals no information about x or y while the non-differential term unlocks the code to reveal both. The validity of differentially encoded tuples may be trivially confirmed given knowledge of both terms.

We define the differential hash operators,

$$\begin{aligned} \Delta(x) &= h(x) \oplus x, \\ \Delta_\pi(x) &= h(x_\pi) \oplus x, \end{aligned} \tag{19.2}$$

where $\pi \in S_n$ for $x \in \{0,1\}^n$ is a permutation over the elements of x. These encode a hash's image and pre-image together while revealing neither assuming hash pre-image resistance. We have the properties,

$$\begin{aligned} h(x) &= \Delta(x) \oplus x, \\ x &= \Delta(x) \oplus h(x). \end{aligned} \tag{19.3}$$

The Δ operator inherits pre-image resistance from $h(\cdot)$. Knowing $\Delta(x)$ alone reveals neither x nor $h(x)$, however additionally knowing x or $h(x)$ enables verification of $\Delta(x)$. Finding x for given $\Delta(x)$ reduces to the pre-image resistance of the hash function $h(\cdot)$.

The non-differentially encoded tuple $\{x, h(x)\}$ allows x to unlock $h(x)$, while $h(x)$ cannot unlock x. The second element reveals $h(x)$ alone, but not x via pre-image resistance. Under the differential encoding,

$$[x, \Delta(x)]_{\oplus} = [x, h(x) \oplus x]_{\oplus}, \tag{19.4}$$

the second element reveals neither x nor $h(x)$, while the first element reveals both, given that $h(x)$ can be efficiently forward-evaluated. Alternately, under the differential encoding,

$$[h(x), \Delta(x)] = [h(x), h(x) \oplus x], \tag{19.5}$$

the non-differential term $h(x)$ affords unlocking the code but does not on its own reveal x via hash pre-image resistance. Under both encodings knowing either x or $h(x)$ alone enables verification.

The differential operator is distributive only over its unhashed components,

$$\begin{aligned} \Delta(x \oplus y) &= h(x \oplus y) \oplus x \oplus y \\ \Delta(x) \oplus \Delta(y) &= h(x) \oplus h(y) \oplus x \oplus y. \end{aligned} \tag{19.6}$$

The symmetric difference between $\Delta(x \oplus y)$ and $\Delta(x) \oplus \Delta(y)$ gives the 'distributor' (equivalent of commutator for distributivity),

$$\Delta(x \oplus y) \oplus \Delta(x) \oplus \Delta(y) = h(x \oplus y) \oplus h(x) \oplus h(y), \tag{19.7}$$

defining the distributivity of Δ operator over the action of $\oplus$.

Standard differential codes are composable,

$$\begin{aligned}&[x, x\oplus y)]_\oplus \oplus [x', x'\oplus y']_\oplus \\ &\sim [x\oplus x', x\oplus y\oplus x'\oplus y']_\oplus .\end{aligned} \tag{19.8}$$

For differential hash codes,

$$\begin{aligned}&[x, \Delta(x)]_\oplus , \\ &[y, \Delta(y)]_\oplus ,\end{aligned} \tag{19.9}$$

we have distinct composition rules,

$$\begin{aligned}&[x\oplus y, \Delta(x\oplus y)]_H , \\ &[x\oplus y, \Delta(x)\oplus\Delta(y))]_\oplus .\end{aligned} \tag{19.10}$$

The $[\cdot,\cdot]_H$ composition is verifiable by hashing the left hand term. The $[\cdot,\cdot]_\oplus$ composition is not hash-verifiable but preserves all differential encoding constraints.

Permutations π are distributive over $\oplus$ but not commutative,

$$\begin{aligned}&\pi(x\oplus y) = \pi(x)\oplus\pi(y), \\ &\pi(x)\oplus y \neq x\oplus\pi(y),\end{aligned} \tag{19.11}$$

whereas $\oplus$ is commutative but not distributive (in general, depending on parity of number of terms under distribution),

$$x\oplus y = y\oplus x. \tag{19.12}$$

- $h(m\oplus s)$ will reveal the private $h(s)$ for chosen $m = \mathbf{0}$.
- $h(m\oplus s\oplus x)$ will reveal the private $h(x)$ for chosen $m = x$ if x is public.
- $h(\pi(m\oplus s)) = h(m_\pi\oplus s_\pi)$ can only reveal $h(s_\pi)$ (not secret) for public π and chosen m, but cannot reveal secret $h(s)$.

XX. ASYMMETRIC CODES

We define key-pairs as,

$$\begin{aligned}\mathtt{sk} &\in \{0,1\}^n, \\ \mathtt{pk} &= \{\mathtt{pk}_\Delta, \mathtt{pk}_\pi\}, \\ \mathtt{pk}_\Delta &= \Delta(\mathtt{sk}), \\ \mathtt{pk}_\pi &\in S_n,\end{aligned} \tag{20.1}$$

where $\mathtt{sk}$ be a secret bit-string, $h(\{0,1\}^n) \to \{0,1\}^n$ an n-bit endomorphic hash function, and $\pi \in S_n$ a permutation on n bits. Since $|S_n| = n!$ encoding π requires $\lceil \log_2(n!)\rceil$ bits. For $n = 256$ we have $\lceil \log_2(n!)\rceil = 1684$ bits.

Since $\mathtt{pk} = \Delta(\mathtt{sk})$ is public both $\mathtt{sk}$ and $h(\mathtt{sk})$ must be private to prevent unlocking the public key, both acting as trapdoors for the differential encoding.

$$[m_\pi\oplus s_\pi, \Delta_\pi(m_\pi\oplus s_\pi)], \tag{20.2}$$

$$\begin{aligned}\Delta_\pi(m_\pi\oplus s_\pi) = \Delta_\pi(m_\pi)\Delta_\pi(s_\pi) \\ \oplus h(m_\pi)\oplus h(s_\pi)\oplus h(m_\pi\oplus s_\pi)\end{aligned} \tag{20.3}$$

XXI. DIGITAL SIGNATURES

To sign message $m \in \{0,1\}^n$ Alice makes public the differentially encoded, signed message $\Delta(m_\pi)$ and signature,

$$\mathtt{sig}_\pi(m) = h(m_\pi\oplus\mathtt{sk}_\pi). \tag{21.1}$$

The signature has the property that when combined with public information it reveals the hash of the message being signed,

$$\mathtt{sig}_\pi(m)\oplus\Delta(m)\oplus\Delta(\mathtt{sk}_\pi) = h(m). \tag{21.2}$$

Employing the modulated public key $\Delta(\mathtt{sk}_\pi)$,

$$\mathtt{sk}_\pi \equiv \mathtt{sk}\oplus h(\mathtt{pk}), \tag{21.3}$$

prevents the signature from revealing the trapdoor $h(\mathtt{sk})$ which unlocks the public key,

$$\begin{aligned}&h(\mathtt{sk})\oplus\Delta(\mathtt{sk}) = \mathtt{sk}, \\ &h(\mathtt{sk}_\pi)\oplus\Delta(\mathtt{sk}) \neq \mathtt{sk}.\end{aligned} \tag{21.4}$$

XXII. ENCRYPTION

For asymmetric encryption we reverse the roles of m and $h(m)$. Bob wishes to send message m to Alice and makes public,

$$\mathtt{enc}(m) = \{\Delta(m), h(m)\}. \tag{22.1}$$

Alice now decrypts using the message hash $h(m)$ to reveal the original message,

$$\Delta(s_\pi)\oplus\Delta(m)\oplus h(m) = m. \tag{22.2}$$

A. Multi-sigs

Signatures are commutative, additive and composable.

$$\mathtt{sig}_\pi(m) = \Delta(s_\pi)\oplus h(m), \tag{22.3}$$

B. Key-establishment & secret sharing

Using the asymmetric encryption protocol, Alice finally communicates the verification hash,

$$\texttt{key} = h(\texttt{sk}\oplus m), \tag{22.4}$$

back to Bob, who also able to verify its validity. The verification hash now provides confirmation of a jointly prepared hash-based random number given by the XOR-salted hash of Alice's secret key and Bob's chosen salt, which cannot be spoofed by either.

XXIII. NOTES

* The hash collision space translates to decoding failure.

Differentially encoded tuples are composable under bitwise XOR,

$$[x,\Delta(x)]\oplus[y,\Delta(y)] = [x\oplus y,\Delta(x)\oplus\Delta(y)], \tag{23.1}$$

via the commutativity of $\oplus$.

Compare above rhs,

$$h(x\oplus y)\oplus x\oplus y = h(x\oplus y)\oplus h(x)\oplus h(y). \tag{23.2}$$

π known by inverting π and multiplying the tuple elements to confirm consistency. For unknown or incorrect π hash verification fails.

The bit permutation operator, $\pi(\cdot)$, is distribute over the bit-wise XOR operator, $\oplus$,

$$\pi(x)\oplus\pi(y) = \pi(x\oplus y). \tag{23.3}$$

Hence differential encoding relationships are preserved under the uniform action of π,

$$[x, x\oplus y] \sim [\pi(x),\pi(x\oplus y)], \tag{23.4}$$

but are in general not preserved under non-uniform action of π,

$$[x, x\oplus y] \not\sim [\pi(x), x\oplus y]. \tag{23.5}$$

We'll employ the shorthand,

$$\pi\circ[x,y] = [\pi(x),\pi(y)]., \tag{23.6}$$

to denote the uniform action of π over a differential code.

While permutations are distributive over $\oplus$ they do not commute through hashes,

$$\pi(h(x)) \neq h(\pi(x)). \tag{23.7}$$

$$\Delta(\pi(x\oplus y)) = h(\pi(x\oplus y))\oplus\pi(x)\oplus\pi(y). \tag{23.8}$$

XXIV. FROM OTHER DOCUMENT

1. Notes

If $\Delta(x) = h(x)\oplus x$ is public information both x and $h(x)$ must be private.

Decrypt m with $h(m)$ requires the locking construction,

$$\begin{aligned}&[h(m), m\oplus h(m)]_\oplus = [h(m),\Delta(m)]_\oplus,\\ &[s\oplus h(m),\Delta(s)\oplus\Delta(m)]_\oplus,\\ &[s\oplus h(m),\Delta(s\oplus h(m))]_\oplus\\ &[s\oplus h(m),\Delta(s)\oplus\Delta(m)\oplus h(s\oplus h(m))\oplus h(m)\oplus h(s)]_\oplus,\end{aligned} \tag{24.1}$$

if $\Delta(m)$ is public.

This does not unlock from known π,

$$[h(\pi(m)),\pi(m)\oplus h(m)]_\oplus \tag{24.2}$$

This unlocks,

$$[h(\pi(s\oplus m)),\Delta(s\oplus m)]_\oplus \tag{24.3}$$

Want to decrypt with,

$$\begin{aligned}h(\pi(s\oplus h(m)) = \Delta(\pi(s\oplus h(m)))\\ \oplus\pi(s)\oplus\pi(h(m)).\end{aligned} \tag{24.4}$$

$$[h(m),\Delta(\pi(m))\oplus\Delta(\pi(s))]_\oplus \tag{24.5}$$

unlocks for known $\Delta(\pi(s))$ (private).

$$\Delta(\pi(m))\oplus\Delta(\pi(s)) = \Delta \tag{24.6}$$

$$\begin{aligned}\texttt{sig} = \Delta(m) = \pi(m)\oplus h(\pi(m))\\ \pi(m) = \texttt{sig}\oplus h(\pi(m))\end{aligned} \tag{24.7}$$

A. Digital signatures

Consider parties Alice and Bob where Alice wants to sign the message $m\in\{0,1\}^n$. Alice prepares the signature,

$$\texttt{sig}(m) = \{\pi(m),\Delta(m),h(\texttt{sk}\oplus m)\}. \tag{24.8}$$

Bob prepares $\pi(\Delta(m))$ and knows $\pi(\Delta(\texttt{sk}))$. We have the relationship,

$$\pi(\texttt{sk}\oplus m) = \pi(\Delta(\texttt{sk})\oplus\Delta(m)\oplus h(m)\oplus(h(\texttt{sk})) \tag{24.9}$$

The encodings,

$$\begin{aligned}&[\mathtt{sk}, \Delta(\mathtt{sk})]_H, [h(\mathtt{sk}), \Delta(\mathtt{sk})]_{\oplus}\\ &[m, \Delta(m)]_H, [h(m), \Delta(m)]_{\oplus}\\ &[m \oplus \mathtt{sk} \oplus h(m) \oplus h(\mathtt{sk}), \Delta(m) \oplus \Delta(\mathtt{sk})]_{\oplus}\end{aligned} \tag{24.10}$$

enable verification by hashing the first term.

The differential code,

$$[\mathtt{sk} \oplus m, \Delta(\mathtt{sk} \oplus m)] \tag{24.11}$$

is verifiable given $h(\mathtt{sk} \oplus m)$, while the permuted code,

$$[\pi(\mathtt{sk} \oplus m), \Delta(\pi(\mathtt{sk} \oplus m))]_H \tag{24.12}$$

is only verifiable with $h(\pi(\mathtt{sk} \oplus m))$.

$$\begin{aligned}&[h(\mathtt{sk} \oplus m), \Delta(\mathtt{sk} \oplus m)]_{\oplus}\\ &[h(\pi(\mathtt{sk} \oplus m)), \Delta(\pi(\mathtt{sk} \oplus m))]_{\oplus}\end{aligned} \tag{24.13}$$

are both verifiable from direct multiplication.

B. Blah

Hence,

$$[h(\pi(\mathtt{sk} \oplus m)), \Delta(\pi(\mathtt{sk} \oplus m))] \tag{24.14}$$

affords direct verification by multiplying $(t_1 \cdot t_2 = h(t_2))$. The second term is equivalent to,

$$\begin{aligned}t_2 &= \Delta(\pi(\mathtt{sk} \oplus m))\\ &= \Delta(\pi(\mathtt{sk})) \oplus \Delta(\pi(m))\\ &\oplus h(\pi(\mathtt{sk} \oplus m)) \oplus h(\pi(\mathtt{sk})) \oplus h(\pi(m))\end{aligned} \tag{24.15}$$

Let signature be,

$$\mathtt{sig} = h(\pi(\mathtt{sk} \oplus m)), \tag{24.16}$$

$$\mathtt{ver} = \Delta(\pi(\mathtt{sk} \oplus m)) \tag{24.17}$$

$$[\mathtt{sig}, \mathtt{ver}] \tag{24.18}$$

XXV. BLAH

$$\begin{aligned}\Delta(\pi(\mathtt{sk} \oplus m)) &= h(\pi(\mathtt{sk} \oplus m)) \oplus \pi(\mathtt{sk} \oplus m)\\ &= h(\pi(\mathtt{sk}) \oplus \pi(m)) \oplus \pi(\mathtt{sk} \oplus m).\\ \Delta(\pi(\mathtt{sk})) \oplus \Delta(\pi(m)) &= \pi(\mathtt{sk}) \oplus \pi(m)\\ &\oplus h(\pi(\mathtt{sk})) \oplus h(\pi(m)).\\ \Delta(\pi(\mathtt{sk} \oplus m)) &= \Delta(\pi(\mathtt{sk})) \oplus \Delta(\pi(m))\\ &\oplus h(\pi(\mathtt{sk} \oplus m)) \oplus h(\pi(\mathtt{sk})) \oplus h(\pi(m))\end{aligned} \tag{25.1}$$

$$\Delta(x \oplus \pi(x)) = h(x \oplus \pi(x)) \oplus x \oplus \pi(x). \tag{25.2}$$

$$\begin{aligned}\Delta(\mathtt{sk} \oplus m \oplus \pi(\mathtt{sk} \oplus m)) &= h(\mathtt{sk} \oplus m \oplus \pi(\mathtt{sk}) \oplus \pi(m))\\ &\mathtt{sk} \oplus m \oplus \pi(\mathtt{sk}) \oplus \pi(m).\end{aligned} \tag{25.3}$$

The differential permuted code,

$$[\pi(\mathtt{sk} \oplus m), \Delta(\pi(\mathtt{sk} \oplus m))] \tag{25.4}$$

is verifiable from,

$$h(\pi(\mathtt{sk} \oplus m)) \tag{25.5}$$

The differential permuted code,

$$[\pi(\mathtt{sk} \oplus m), \Delta(\pi(\mathtt{sk} \oplus m))] \tag{25.6}$$

is verifiable from,

$$\begin{aligned}&h(\pi(\mathtt{sk} \oplus m)) \oplus \pi(\mathtt{sk} \oplus m) = bob\\ &\oplus h(\pi(\mathtt{sk} \oplus m)) \oplus h(\pi(\mathtt{sk})) \oplus h(\pi(m))\\ &=\end{aligned} \tag{25.7}$$

While,

$$\begin{aligned}&[\mathtt{sk} \oplus m, \Delta(\mathtt{sk} \oplus m)]\\ &= [\mathtt{sk} \oplus m, \Delta(\mathtt{sk}) \oplus \Delta(m)\\ &\oplus h(\mathtt{sk} \oplus m) \oplus h(\mathtt{sk}) \oplus h(m)],\end{aligned} \tag{25.8}$$

is verifiable from $h(\mathtt{sk} \oplus m)$.

Bob's test passes if,

$$\pi(\mathtt{sk} \oplus m) = \Delta(\pi(\mathtt{sk})) \oplus \Delta(\pi(m)) \oplus h(\pi(\mathtt{sk})) \oplus h(\pi(m)) \tag{25.9}$$

Bob is not allowed to know $h(\mathtt{sk})$.

Similarly,

$$\begin{aligned}&[\mathtt{sk} \oplus m, \Delta(\mathtt{sk} \oplus m)]\\ &= [\mathtt{sk} \oplus m, h(\mathtt{sk} \oplus m) \oplus h(\mathtt{sk}) \oplus h(m)],\end{aligned} \tag{25.10}$$

are verifiable from hashing the first term, while,

$$\begin{aligned}&[\pi(\mathtt{sk} \oplus m), \Delta(\mathtt{sk} \oplus m)]\\ &= [\pi(\mathtt{sk} \oplus m), h(\mathtt{sk} \oplus m) \oplus h(\mathtt{sk}) \oplus h(m)],\end{aligned} \tag{25.11}$$

are verifiable by hashing the unpermuted first term. Equivalently,

$$\begin{aligned}&[\pi(\mathtt{sk} \oplus m), \Delta(\mathtt{sk} \oplus m)]\\ &[\pi(\mathtt{sk} \oplus m), \Delta(\mathtt{sk}) \oplus \Delta(m) \oplus h(\mathtt{sk} \oplus m) \oplus h(\mathtt{sk}) \oplus h(m)]\end{aligned} \tag{25.12}$$

is verifiable by hashing the unpermuted first term.

Using,

$$\Delta(\mathtt{sk})\oplus\Delta(m)=\Delta(\mathtt{sk}\oplus m)\oplus h(\mathtt{sk}\oplus m)\oplus h(\mathtt{sk})\oplus h(m), \tag{25.13}$$

Ver:

$$\begin{gathered}\{\pi(m)\oplus\pi(\mathtt{sk}),\Delta(m)\oplus\Delta(\mathtt{sk})\}\\ \{\pi(m\oplus\mathtt{sk}),\Delta(m)\oplus\Delta(\mathtt{sk})\}\end{gathered} \tag{25.14}$$

We have the relationships,

$$\begin{aligned}\Delta(\mathtt{sk})\oplus\Delta(m)&=\mathtt{sk}\oplus m\oplus h(\mathtt{sk})\oplus h(m),\\ \Delta(\mathtt{sk}\oplus m)&=h(\mathtt{sk}\oplus m)\oplus\mathtt{sk}\oplus m,\end{aligned} \tag{25.15}$$

Under composition we obtain,

$$\Delta(\mathtt{sk}\oplus m)\oplus\Delta(\mathtt{sk})\oplus\Delta(m)=h(\mathtt{sk}\oplus m)\oplus h(\mathtt{sk})\oplus h(m). \tag{25.16}$$

The differential encoding,

$$[h(\mathtt{sk}\oplus m),\Delta(\mathtt{sk}\oplus m)], \tag{25.17}$$

Consider the permuted differential code,

$$[\pi(\mathtt{sk}\oplus m),\pi(\Delta(\mathtt{sk}))\oplus\pi(\Delta(m))]. \tag{25.18}$$

we equivalently obtain,

$$[\Delta(\mathtt{sk}\oplus m)\oplus h(\mathtt{sk}\oplus m),\Delta(\mathtt{sk})\oplus\Delta(m)]. \tag{25.19}$$

Therefore if Alice provides the signature,

$$\mathtt{sig}(m)=[\pi(\mathtt{sk}\oplus m),h(\mathtt{sk}\oplus m)], \tag{25.20}$$

Bob is able to verify via,

$$[\mathtt{sig}(m),\Delta(\mathtt{sk})\oplus\Delta(m)]. \tag{25.21}$$

REFERENCES