

Universal hash-based post-quantum cryptography

Peter P. Rohde^{1,2,3,4}

¹*BTQ Technologies, 16-104 555 Burrard Street, Vancouver BC, V7X 1M8 Canada*

²*Center for Engineered Quantum Systems, School of Mathematical & Physical Sciences, Macquarie University, NSW 2109, Australia*

³*Centre of Excellence in Engineered Quantum Systems, Brisbane, Australia*

⁴*Hearne Institute for Theoretical Physics, Department of Physics & Astronomy, Louisiana State University, Baton Rouge LA, United States*

I. DIFFERENTIAL HASH CODES

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

$$[x, x \oplus y]_{\oplus}, \quad (1.1)$$

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,

$$\begin{aligned} \Delta(x) &= h(x) \oplus x, \\ \Delta_{\pi}(x) &= h(x_{\pi}) \oplus x, \end{aligned} \quad (1.2)$$

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,

$$\begin{aligned} h(x) &= \Delta(x) \oplus x, \\ x &= \Delta(x) \oplus h(x). \end{aligned} \quad (1.3)$$

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,

$$[x, \Delta(x)]_{\oplus} = [x, h(x) \oplus x]_{\oplus}, \quad (1.4)$$

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,

$$[h(x), \Delta(x)] = [h(x), h(x) \oplus x], \quad (1.5)$$

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,

$$\begin{aligned} \Delta(x \oplus y) &= h(x \oplus y) \oplus x \oplus y \\ \Delta(x) \oplus \Delta(y) &= h(x) \oplus h(y) \oplus x \oplus y. \end{aligned} \quad (1.6)$$

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

$$\Delta(x \oplus y) \oplus \Delta(x) \oplus \Delta(y) = h(x \oplus y) \oplus h(x) \oplus h(y), \quad (1.7)$$

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

Standard differential codes are composable,

$$\begin{aligned} [x, x \oplus y]_{\oplus} \oplus [x', x' \oplus y']_{\oplus} \\ \sim [x \oplus x', x \oplus y \oplus x' \oplus y']_{\oplus}. \end{aligned} \quad (1.8)$$

For differential hash codes,

$$\begin{aligned} [x, \Delta(x)]_{\oplus}, \\ [y, \Delta(y)]_{\oplus}, \end{aligned} \quad (1.9)$$

we have distinct composition rules,

$$\begin{aligned} [x \oplus y, \Delta(x \oplus y)]_H, \\ [x \oplus y, \Delta(x) \oplus \Delta(y)]_{\oplus}. \end{aligned} \quad (1.10)$$

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,

$$\begin{aligned} \pi(x \oplus y) &= \pi(x) \oplus \pi(y), \\ \pi(x) \oplus y &\neq x \oplus \pi(y), \end{aligned} \quad (1.11)$$

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

$$x \oplus y = y \oplus x. \quad (1.12)$$

- $h(m \oplus s)$ 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 $h(s)$.

II. ASYMMETRIC CODES

We define key-pairs as,

$$\begin{aligned} \mathbf{sk} &\in \{0, 1\}^n, \\ \mathbf{pk} &= \{\mathbf{pk}_\Delta, \mathbf{pk}_\pi\}, \\ \mathbf{pk}_\Delta &= \Delta(\mathbf{sk}), \\ \mathbf{pk}_\pi &\in S_n, \end{aligned} \quad (2.1)$$

where \mathbf{sk} be a secret bit-string, $h(\{0, 1\}^n) \rightarrow \{0, 1\}^n$ 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 $\mathbf{pk} = \Delta(\mathbf{sk})$ is public both \mathbf{sk} and $h(\mathbf{sk})$ must be private to prevent unlocking the public key, both acting as trapdoors for the differential encoding.

$$[m_\pi \oplus s_\pi, \Delta_\pi(m_\pi \oplus s_\pi)], \quad (2.2)$$

$$\begin{aligned} \Delta_\pi(m_\pi \oplus s_\pi) &= \Delta_\pi(m_\pi) \Delta_\pi(s_\pi) \\ &\oplus h(m_\pi) \oplus h(s_\pi) \oplus h(m_\pi \oplus s_\pi) \end{aligned} \quad (2.3)$$

III. DIGITAL SIGNATURES

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

$$\mathbf{sig}_\pi(m) = h(m_\pi \oplus \mathbf{sk}_\pi). \quad (3.1)$$

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

$$\mathbf{sig}_\pi(m) \oplus \Delta(m) \oplus \Delta(\mathbf{sk}_\pi) = h(m). \quad (3.2)$$

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

$$\mathbf{sk}_\pi \equiv \mathbf{sk} \oplus h(\mathbf{pk}), \quad (3.3)$$

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

$$\begin{aligned} h(\mathbf{sk}) \oplus \Delta(\mathbf{sk}) &= \mathbf{sk}, \\ h(\mathbf{sk}_\pi) \oplus \Delta(\mathbf{sk}) &\neq \mathbf{sk}. \end{aligned} \quad (3.4)$$

IV. ENCRYPTION

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

$$\mathbf{enc}(m) = \{\Delta(m), h(m)\}. \quad (4.1)$$

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

$$\Delta(s_\pi) \oplus \Delta(m) \oplus h(m) = m. \quad (4.2)$$

A. Multi-sigs

Signatures are commutative, additive and composable.

$$\mathbf{sig}_\pi(m) = \Delta(s_\pi) \oplus h(m), \quad (4.3)$$

B. Key-establishment & secret sharing

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

$$\mathbf{key} = h(\mathbf{sk} \oplus m), \quad (4.4)$$

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.

V. NOTES

* The hash collision space translates to decoding failure.

Differentially encoded tuples are composable under bit-wise XOR,

$$[x, \Delta(x)] \oplus [y, \Delta(y)] = [x \oplus y, \Delta(x) \oplus \Delta(y)], \quad (5.1)$$

via the commutativity of \oplus .

Compare above rhs,

$$h(x \oplus y) \oplus x \oplus y = h(x \oplus y) \oplus h(x) \oplus h(y). \quad (5.2)$$

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 ,

$$\pi(x) \oplus \pi(y) = \pi(x \oplus y). \quad (5.3)$$

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

$$[x, x \oplus y] \sim [\pi(x), \pi(x \oplus y)], \quad (5.4)$$

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

$$[x, x \oplus y] \not\sim [\pi(x), x \oplus y]. \quad (5.5)$$

We'll employ the shorthand,

$$\pi \circ [x, y] = [\pi(x), \pi(y)]., \quad (5.6)$$

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

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

$$\pi(h(x)) \neq h(\pi(x)). \quad (5.7)$$

$$\Delta(\pi(x \oplus y)) = h(\pi(x \oplus y)) \oplus \pi(x) \oplus \pi(y). \quad (5.8)$$

REFERENCES