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,"
  * one-way permutation.
  * Trapdoor function.

- 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
- Inverting braids $\begin{bmatrix} x \\ x \end{bmatrix}_{\pi}^{\bar{\pi}} \to x$ is equivalent to the Conjugacy Problem for Permutations (CPP), known to be NP-complete. For braid codes complexity is maximised for the balanced pre-image space, $\mathrm{wt}(x) = n/2$. Nb: requires double braiding.
- Permutation space is larger than bitstring space. Hence necessarily automorphisms.
- Invertible hash functions related by permutations in mutual preimage space.
- As morphisms both permutations and XOR masks preserve bijectivity. Symmetrise the hash model via $h(x) = h^{-1}(x)$. This is bijective as is $\pi \circ h(X) \oplus c = h^{-1}(x)$.
- Hash function model has three orders of braining in each direction. Every braid order is pre-image protected against the next. Braid $m \oplus s$ against common π. This equates to a difference term c' against braided s. Sending difference term to another party holding s allows reconstruction. This is made asymmetric by braiding against particular braid orders. If Alice shares only a braid, lower levels cannot be inferred. But the difference terms can be evaluated with different combinations of levels. Allows asymmetric encoding in either direction.
- Permutations and XOR masks preserve bijectivity. If a function $f : X \to X$ is bijective then so is $f : X \to \pi \circ X \oplus c$.

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 pre-image 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. BRAID CODE SUMMARY

Asymmetric encryption $(A \to B : m)$:

$$\begin{aligned}
\mathtt{sig}_A &: \begin{bmatrix} h(m \oplus p) \\ m \end{bmatrix}_{\pi}^{\bar{\pi}}, \\
\mathtt{key}_B &: \begin{bmatrix} s \\ p \end{bmatrix}_{\pi}^{\bar{\pi}}, \\
\mathtt{dec}_B &: \begin{bmatrix} h(m \oplus p) \\ m \end{bmatrix}_{\pi}^{\bar{\pi}} \oplus \begin{bmatrix} s \\ p \end{bmatrix}_{\pi}^{\bar{\pi}} = \begin{bmatrix} h(m \oplus p) \oplus s \\ m \oplus p \end{bmatrix}_{\pi}^{\bar{\pi}},
\end{aligned} \tag{3.1}$$

where knowledge of $\{s, p, h(m), \pi\}$ affords the decryption stage.

Digital signature for m:

$$\begin{aligned}
\mathtt{sig}_A &: h(m \oplus p), \\
\mathtt{key}_B &: \begin{bmatrix} s \\ p \end{bmatrix}_{\pi}^{\bar{\pi}}, \\
\mathtt{dec}_B &: \begin{bmatrix} m \\ h(m) \end{bmatrix}_{\pi}^{\bar{\pi}} \oplus \begin{bmatrix} s \\ p \end{bmatrix}_{\pi}^{\bar{\pi}} = \begin{bmatrix} m \oplus s \\ h(m) \oplus p \end{bmatrix}_{\pi}^{\bar{\pi}}.
\end{aligned} \tag{3.2}$$

IV. ENTROPY ALGEBRA

Considering the abelian binary group of length n bitstrings under bit-wise XOR,

$$G_{\mathrm{B}} = (\mathbb{Z}_2^n, \oplus) \sim (\{0,1\}^n, \oplus), \tag{4.1}$$

we consider subgroups generated by finite alphabets of k randomly sampled letters,

$$\begin{aligned}
&\mathcal{X} : \Sigma \in \mathbb{Z}_2^n,\ k = |\Sigma|, \\
&G_\Sigma = \langle \Sigma \rangle \subseteq G_{\mathrm{B}},
\end{aligned} \tag{4.2}$$

where individual generators compose under $(\mathbb{Z}_2, \oplus)$,

$$g_i \oplus g_i = e \sim \mathbf{0}, \tag{4.3}$$

and the identity element is the zero vector. Entropy groups have group generators representing independent

random variables,

$$G_{\rm E} = \langle \mathcal{X}_{i\in\{1,k\}} \rangle \sim \mathbb{Z}_2^k,$$
$$|G_{\rm E}| = 2^k, \tag{4.4}$$

where group elements comprise all binary combinations of group generators.

Group elements represent n-bit information objects given by the XOR of their constituent generators. Independent maximum entropy letters,

$$H(\mathcal{X}) = 1, \tag{4.5}$$

where $H(\cdot)$ denotes Shannon entropy, exhibit no mutual information,

$$I(\mathcal{X}_i, \mathcal{X}_j) = \delta_{i,j}, \tag{4.6}$$

and are statistically distinguishable, thereby constituting distinct letters in the algebra. Zero-entropy letters reduce to constants,

$$H(\mathcal{X} \to c) = 0. \tag{4.7}$$

Information theoretically, for maximum entropy random variables the entropy group generators are distinct,

$$\begin{aligned} G_{\rm E} &= \langle \mathcal{X}_i, \mathcal{X}_j \rangle, \\ &= \{e, \mathcal{X}_i, \mathcal{X}_j, \mathcal{X}_i \oplus \mathcal{X}_j\} \\ &= \langle \mathcal{X}_i \rangle \times \langle \mathcal{X}_j \rangle \\ &\sim \mathbb{Z}_2 \times \mathbb{Z}_2, \end{aligned} \tag{4.8}$$

whereas zero-entropy random variables have no information-theoretic content, reducing the action of the entropy group to the trivial group,

$$\begin{aligned} &\phi : \mathcal{X} \to e, \\ &\tilde{G}_{\rm E} = \{e\} \sim \mathbf{1}. \end{aligned} \tag{4.9}$$

defining a group homomorphism mapping entropic code groups to non-entropic ones,

$$\begin{aligned} &\phi : G_{\rm C} \to \tilde{G}_{\rm C}, \\ &\ker(\tilde{G}_{\rm C}) = G_{\rm E}, \\ &\tilde{G}_{\rm C} = G_{\rm C} \,/\, \ker(\phi) = G_{\rm C}/G_{\rm E}. \end{aligned} \tag{4.10}$$

Code groups may contain both entropic ($\mathcal{X}$) and non-entropic generators (x) and group elements in general comprise letters from both ($\mathcal{X} \oplus x$). Non-entropic group elements expose the joint parity of their constituent group generators (x), whereas entropic elements ($\mathcal{X} \oplus x$) do not. The entropic and non-entropic subgroups partition code groups into a direct product (sum under $\oplus$) decomposition,

$$G = G_{\mathcal{X}} \times \tilde{G}_{\mathcal{X}}, \tag{4.11}$$

representing accessible and inaccessible information. Elements of the entropic subgroup $G_{\mathcal{X}}$ information-theoretically protect their non-entropic terms.

Different parties in a multi-party communications protocol in general possess different generating sets defining their subjective code groups. In the context of an encryption protocol the goal is for communicating parties to establish private code groups exposing encrypted data, while ensuring public code groups exposes only protected entropic objects.

A. Symmetric encryption

Consider an entropy code group comprising two random variables $\{\mathcal{X}_i, \mathcal{X}_j\}$, secrets $\{s, p\}$, and message m, defining the group generators,

$$G_{\rm C} = \langle s, p, m, \mathcal{X}_i, \mathcal{X}_j \rangle \sim \mathbb{Z}_2^5. \tag{4.12}$$

with $|G_{\rm C}| = 2^5$ group elements.

Assume Alice and Bob share secrets $\{s, p\}$. Let Alice prepare two random variables $\{\mathcal{X}_i, \mathcal{X}_j\}$ and message m. Both parties respectively publicly share,

$$\begin{aligned} &B \to A : \{s \oplus \mathcal{X}_j\}. A \to B : && \{s \oplus \mathcal{X}_i^A, \mathcal{X} \\ &B \to A : \{s \oplus \mathcal{X}_i^B, \mathcal{X}_i^B \oplus \mathcal{X}_j^B\}. A \to B : && \texttt{pub} = \{s \oplus \mathcal{X}_i, \mathcal{X}_i \oplus \mathcal{X}_j, p \\ &B \to A : \quad \texttt{pub} = \{s \oplus \mathcal{X}_i, \mathcal{X}_i \oplus \mathcal{X}_j, p \oplus m \oplus \mathcal{X}_j\}. \end{aligned} \tag{4.13}$$

As s is known to both Alice and Bob the both possess the private code group,

$$G = \{s, \mathcal{X}_i, \mathcal{X}_j, s \oplus \mathcal{X}_i, s \oplus \mathcal{X}_j, \mathcal{X}_i \oplus \mathcal{X}_j, s \oplus \mathcal{X}_i \oplus \mathcal{X}_j\}.$$

The group generated by public information is,

$$\begin{aligned} &= \{s \oplus \mathcal{X}_i, s \oplus \mathcal{X}_j, \mathcal{X}_i \oplus \mathcal{X}_j, \\ &\quad p \oplus m \oplus \mathcal{X}_i, p \oplus m \oplus \mathcal{X}_j, \\ &\quad s \oplus p \oplus m \oplus \mathcal{X}_i, s \oplus p \oplus m \oplus \mathcal{X}_j\}, G_{\rm C}^{\rm pub} \quad = \langle s \oplus \mathcal{X}_i, \mathcal{X}_i \oplus \mathcal{X}_j, p \oplus m \oplus \mathcal{X}_j \rangle, \\ &= \{s \oplus \mathcal{X}_i, s \oplus \mathcal{X}_j, \mathcal{X}_i \oplus \mathcal{X}_j, \\ &\quad p \oplus m \oplus \mathcal{X}_i, p \oplus m \oplus \mathcal{X}_j, \\ &\quad s \oplus p \oplus m \oplus \mathcal{X}_i, s \oplus p \oplus m \oplus \mathcal{X}_j\}, \end{aligned} \tag{4.14}$$

which does not reveal any non-entropic group elements.

Alice now sends encoded message m,

$$A \to B : \{m \oplus\}. \tag{4.15}$$

while the group defined over private group generators comprises the full generating set of the code group,

$$G_{\rm priv} = G_{\rm enc}. \tag{4.16}$$

The set of a group's elements corresponds to the space of available information objects. Group elements containing random variables are themselves random variables. The group element,

$$g = x \oplus \mathcal{X}, \tag{4.17}$$

has entropy,

$$H(g) \geq \max[H(x), H(\mathcal{X})], \tag{4.18}$$

for arbitrary x, and for $H(\mathcal{X}) = 1$ the object g is a maximum entropy random variable revealing no information about x,

$$I(x, g) \leq 1 - H(\mathcal{X}), \tag{4.19}$$

where $I(x, g)$ is measured per-bit. In the zero-entropy regime,

$$H(\mathcal{X}) = 0, \tag{4.20}$$

we have,

$$x \oplus \mathcal{X} \to x \oplus c, \tag{4.21}$$

for constant c under which variable x is information-theoretically invariant with perfect mutual information,

$$I(x, g) = H(x). \tag{4.22}$$

The respective zero-entropy private and public groups are,

$$\begin{aligned}
\tilde{S}_{\text{priv}} &= \{s, p, p \oplus m\}, \\
\tilde{G}_{\text{priv}} &= \langle \tilde{S}_{\text{priv}} \rangle \\
&= \{s, p, m, s \oplus p, s \oplus m, p \oplus m, s \oplus p \oplus m\}, \\
\tilde{S}_{\text{pub}} &= \{s, p \oplus m\}, \\
\tilde{G}_{\text{pub}} &= \langle \tilde{S}_{\text{pub}} \rangle \\
&= \{s, p \oplus m, s \oplus p \oplus m\}.
\end{aligned} \tag{4.23}$$

The zero-entropy public group contains group element $g = p \oplus m$. As s and p are reusable and static they exhibit no entropy,

$$H(\mathcal{X}_s) = H(\mathcal{X}_p) = 0, \tag{4.24}$$

and the non-entropic group homomorphism reveals the message,

$$\phi : \tilde{G}_{\text{pub}} \to \langle m \rangle. \tag{4.25}$$

Equivalently, treating private key p as a random variable, $g = m \oplus \mathcal{X}_p$ takes the form of a one-time-pad cipher. If p is an infinite maximum entropy stream we have,

$$\begin{aligned}
H(\mathcal{X}_p) &= 1, \\
H(m \oplus \mathcal{X}_p) &= 1.
\end{aligned} \tag{4.26}$$

For reused p we have,

$$\begin{aligned}
H(\mathcal{X}_p) &= 0, \\
\phi : g &\to m.
\end{aligned} \tag{4.27}$$

In the zero-entropy case the scheme reduces to a one-time-pad with reused private key, revealing as plaintext message parities,

$$(p \oplus m_1) \oplus (p \oplus m_2) = m_1 \oplus m_2. \tag{4.28}$$

With increasing messages or a single chosen plaintext this affords decorrelation of private information. Only when p is chosen independently for every m do we achieve information theoretic security.

Conceptually, the scheme uses $\{s, p\}$ to prove parity constraints against other variables with a hidden parity constraint necessary to complete the proof. Against maximum entropy random variables parity proofs act as zero-knowledge proofs (ZKPs) and $\{s, p\}$ are reusable without information leakage so long as they remain statistically distinguishable. Proofs against known variables have no hidden parity revealing the solution.

The secret sharing of entropy achieved by quantum key distribution (QKD) is via the hidden parity constraint imposed by shared entanglement, enforcing measurement correlations in random outcomes. Entropy derives from Heisenberg uncertainty while entanglement-enforced hidden parity ensures mutual outcome. Here entropy is derived client side with hidden parity constraints imposed by the group algebra.

Notes:

* Probability of letter collisions exhibits birthday scaling,

$$P_{\text{C}} = O(k^2 2^{-n}). \tag{4.29}$$

Ensuring $P_{\text{C}} < \delta$ requires space overhead,

$$n = O(\log(k^2/\delta)). \tag{4.30}$$

* Initial round requires two random variables. Subsequent rounds require only one additional random variable per block, sliding against the previous one. The parity constraint is differentially encoded against the parity between s and p, creating a perpetually evolving parity constraint. The first random variable can be interpreted as a key exchange round, followed encryption rounds. Hence the protocol has a fixed 2-fold multiplicative communications overhead with an extra one-off overhead during initial the key exchange round.

* Described above is unidirectional entropy exchange where security is bounded by the entropy of the sender. Performing the protocol in both directions and utilising their joint entropy under XOR ensures at least the entropy of either party, affording a communications channel with security satisfying both. The entropy group remains closed under multiple rounds and the additional group generators do violate the security of the group.

* This can be generalised to arbitrary numbers of parties over a fully connected network with $O(n^2)$ independent two-way exchanges. This enables multi-party key

agreement, where the collective serves as its own entropy oracle.

* From a multi-party entropy exchange every subset of the entropic group generators forms its own secure entropic subgroup reflecting the respective complete subgraph. Every entropic subgroup is secure against all others.

B. Universal homomorphic computing

For an m-bit computation we entropically encode each bit with a unique random variable where the entropic generators are secret,

$$\begin{aligned}\bar{G}_{\mathcal{X}} &= \langle b_i \rangle, \\ G_{\mathcal{X}} &= \langle \mathcal{X}_i \oplus b_i \rangle, \\ & b_i \in \{0,1\},\, i \in \{1,m\}. \end{aligned} \tag{4.31}$$

The respective group elements from both subgroups forming the computational subgroup form an injective group homomorphism under the information-theoretic action $\phi : \mathcal{X} \to c$,

$$\phi : G_{\mathcal{X}} \hookrightarrow \bar{G}_{\mathcal{X}}. \tag{4.32}$$

A universal gate set may be implemented via the NOT and CNOT (reversible XOR) gates,

$$\begin{aligned}\mathrm{NOT}_i : b_i \to \bar{b}_i \equiv b_i \to b_i \oplus \mathbf{1}, \\ \phi : \mathcal{X}_i \oplus b_i \to \mathcal{X}_i \oplus b_i \oplus \mathbf{1},\end{aligned} \tag{4.33}$$

$$\begin{aligned}\mathrm{CNOT}_{i,j} : (b_i, b_j) \to (b_i \oplus b_j, b_j), \\ \phi : (s \oplus \mathcal{X}_i \oplus b_i, p \oplus \mathcal{X}_j \oplus b_j) \\ \to (s \oplus p \oplus \mathcal{X}_i \oplus \mathcal{X}_j \oplus b_i \oplus b_j, p \oplus \mathcal{X}_j \oplus b_j),\end{aligned}$$

with public,

$$\mathtt{pub} : \{s \oplus \mathcal{X}_j\}, \tag{4.34}$$

allowing the delegate to perform the $(p \oplus \mathcal{X}_i, \mathbf{0})$ correction to obtain,

$$(\mathcal{X}_i \oplus \mathcal{X}_j, \mathbf{0}) \times (s \oplus \mathcal{X}_i \oplus b_i, s \oplus \mathcal{X}_j \oplus b_j) \to \tag{4.35}$$

* FIX THIS. Use conjugate parity differentials.

* Use dual-rail complementary logical encoding, $b \to (b, \bar{b})$. This preserves parity under exchange $(\bar{b}, b)$ or joint complement $(a, b) \to (\bar{a}, \bar{b})$ operations.

* Construct a CNOT over a four-bit (2 dual-rail) space which is parity preserving under logical operations, thereby revealing no parity correlations.

* Define logical operations over of parity preserving primitives.

* Client provides server with matrix of entropy correlations with coefficients $M_{i,j} = \mathcal{X}_i \oplus \mathcal{X}_j$. These operators act as swaps when acting on entropic group generators, $(\mathcal{X}_i \oplus \mathcal{X}_j) \oplus \mathcal{X}_i = \mathcal{X}_j$, but do not reveal the generators themselves, preserving closure of protected subgroups.

* Turn homomorphic scheme into blind computing scheme by encoding logical operators under hidden parity constraints.

* Scheme has constant 2-fold multiplicative overhead (space and computational) associated with dual-rail encoding.

* Model: Arbitrary number of bits and a single CNOT acting on fixed bits. Permutation matrices reroute bits to the CNOT to enable universality. Permutation matrices are hidden in the entropy correlation matrix, $M' = P \cdot M \cdot P^{-1}$. Choose the M' matrix to refresh entropic variables so that reuse does not reveal routing information. This has $O(m^2 d)$ overhead for an m-bit circuit with depth d.

C. Non-abelian entropy injection

Entropy injection from source $\tilde{\mathcal{X}}$ into $\mathcal{X}$ implemented via,

$$\mathcal{X}_{i+1} = \begin{bmatrix} \tilde{\mathcal{X}}_{i+1} \\ \tilde{\mathcal{X}_{i+1}} \end{bmatrix}_{\pi}^{\bar{\pi}} \oplus \mathcal{X}_i. \tag{4.36}$$

has entropy,

$$H(\mathcal{X}_{i+1}) \geq \max[H(\mathcal{X}_i), H(\tilde{\mathcal{X}}_i)], \tag{4.37}$$

and is a non-abelian primitive under the assumption of the pre-image resistance of random braids. This prevents compromised sources from choosing,

$$\tilde{\mathcal{X}}_{i+1} = \mathcal{X}_i, \tag{4.38}$$

and eliminating entropy,

$$\mathcal{X}_{i+1} = \mathbf{0}, \tag{4.39}$$

enabling cryptographically secure entropy addition.

V. HASH ALGEBRA

A. Differential hash codes

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

$$[x, x \oplus y]_{\oplus}, \tag{5.1}$$

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,

$$\begin{aligned}\Delta(x) &= h(x) \oplus x, \\ \Delta_{\pi}(x) &= h(x_{\pi}) \oplus x,\end{aligned} \tag{5.2}$$

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,

$$\begin{aligned} h(x) &= \Delta(x) \oplus x, \\ x &= \Delta(x) \oplus h(x). \end{aligned} \tag{5.3}$$

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,

$$[x, \Delta(x)]_{\oplus} = [x, h(x) \oplus x]_{\oplus}, \tag{5.4}$$

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,

$$[h(x), \Delta(x)] = [h(x), h(x) \oplus x], \tag{5.5}$$

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,

$$\begin{aligned} \Delta(x \oplus y) &= h(x \oplus y) \oplus x \oplus y \\ \Delta(x) \oplus \Delta(y) &= h(x) \oplus h(y) \oplus x \oplus y. \end{aligned} \tag{5.6}$$

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

$$\Delta(x \oplus y) \oplus \Delta(x) \oplus \Delta(y) = h(x \oplus y) \oplus h(x) \oplus h(y), \tag{5.7}$$

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

Standard differential codes are composable,

$$\begin{aligned} &[x, x \oplus y)]_{\oplus} \oplus [x', x' \oplus y']_{\oplus} \\ &\sim [x \oplus x', x \oplus y \oplus x' \oplus y']_{\oplus}. \end{aligned} \tag{5.8}$$

For differential hash codes,

$$\begin{aligned} &[x, \Delta(x)]_{\oplus}, \\ &[y, \Delta(y)]_{\oplus}, \end{aligned} \tag{5.9}$$

we have distinct composition rules,

$$\begin{aligned} &[x \oplus y, \Delta(x \oplus y)]_H, \\ &[x \oplus y, \Delta(x) \oplus \Delta(y))]_{\oplus}. \end{aligned} \tag{5.10}$$

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,

$$\begin{aligned} \pi(x \oplus y) &= \pi(x) \oplus \pi(y), \\ \pi(x) \oplus y &\neq x \oplus \pi(y), \end{aligned} \tag{5.11}$$

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

$$x \oplus y = y \oplus x. \tag{5.12}$$

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

VI. BRAID CODE IDENTITIES

Following from the binary identities,

$$\begin{aligned} \bar{x} &= x \oplus \mathbf{1}, \\ \overline{x \oplus y} &= x \oplus \bar{y} = \bar{x} \oplus y. \end{aligned} \tag{6.1}$$

$$\begin{aligned} \Delta_\pi(x) &\equiv \begin{bmatrix} x \\ x \end{bmatrix}_{\pi}^{\bar{\pi}}, \\ [\Delta_\pi \circ \Delta_\pi](x) &= [x_\pi \cdot x_{\pi^{-1}}]_\pi \cdot [x_\pi \cdot x_{\pi^{-1}}]_{\pi^{-1}} \\ &= x_{\pi^2} \cdot x \cdot x \cdot x_{\pi^{-2}} \\ &= x_{\pi^2} \cdot x_{\pi^{-2}} \\ &= \begin{bmatrix} x \\ x \end{bmatrix}_{\pi^2}^{\bar{\pi}^2}, \end{aligned} \tag{6.2}$$

where $x_\pi \equiv \pi \circ x$ and $\bar{\pi} \equiv \pi^{-1}$.

Braid flip:

$$\begin{bmatrix} x \\ y \end{bmatrix}_{\pi}^{\bar{\pi}} \oplus \begin{bmatrix} y \\ x \end{bmatrix}_{\pi}^{\bar{\pi}} = \begin{bmatrix} x \\ x \end{bmatrix}_{\pi}^{\bar{\pi}} \oplus \begin{bmatrix} y \\ y \end{bmatrix}_{\pi}^{\bar{\pi}}. \tag{6.3}$$

Braid complement:

$$\begin{aligned} \overline{\begin{bmatrix} x \\ y \end{bmatrix}_{\pi}^{\bar{\pi}}} &= \begin{bmatrix} \bar{x} \\ y \end{bmatrix}_{\pi}^{\bar{\pi}} = \begin{bmatrix} x \\ \bar{y} \end{bmatrix}_{\pi}^{\bar{\pi}} \\ \begin{bmatrix} x \\ y \end{bmatrix}_{\pi}^{\bar{\pi}} \oplus \begin{bmatrix} \mathbf{1} \\ \mathbf{0} \end{bmatrix}_{\pi}^{\bar{\pi}} &= \begin{bmatrix} x \\ y \end{bmatrix}_{\pi}^{\bar{\pi}} \oplus \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \end{bmatrix}_{\pi}^{\bar{\pi}}. \end{aligned} \tag{6.4}$$

Braid group:

$$e = \begin{bmatrix}\mathbf{0}\\ \mathbf{0}\end{bmatrix}_{\pi}^{\bar{\pi}} = \begin{bmatrix}\mathbf{1}\\ \mathbf{1}\end{bmatrix}_{\pi}^{\bar{\pi}},$$
$$g^{-1} = \bar{g} = g \circ \begin{bmatrix}\mathbf{1}\\ \mathbf{0}\end{bmatrix}_{\pi}^{\bar{\pi}} = g \circ \begin{bmatrix}\mathbf{0}\\ \mathbf{1}\end{bmatrix}_{\pi}^{\bar{\pi}}. \tag{6.5}$$

Braid unlocking:

$$\begin{bmatrix}x\\ y\end{bmatrix}_{\pi}^{\bar{\pi}} \oplus x_{\pi^{-1}} = y_{\pi},$$
$$\begin{bmatrix}x\\ y\end{bmatrix}_{\pi}^{\bar{\pi}} \oplus y_{\pi} = x_{\pi^{-1}}. \tag{6.6}$$

Braid permutations:

$$\pi \circ \begin{bmatrix}x\\ y\end{bmatrix}_{\pi}^{\bar{\pi}} = \begin{bmatrix}x\\ y\end{bmatrix}_{\pi^2}^{e}. \tag{6.7}$$

Rebraiding:

$$\Delta_{\pi} \circ \begin{bmatrix}x\\ y\end{bmatrix}_{\pi}^{\bar{\pi}} = \begin{bmatrix}x\\ y\end{bmatrix}_{\pi^2}^{\bar{\pi}^2} \tag{6.8}$$

* Show identities.

Parity exchange:

$$\begin{bmatrix}x\\ x\end{bmatrix}_{\pi}^{\bar{\pi}} = \begin{bmatrix}y\\ y\end{bmatrix}_{\pi}^{\bar{\pi}} \oplus [x \oplus y]_{\pi},$$
$$\begin{bmatrix}y\\ y\end{bmatrix}_{\pi}^{\bar{\pi}} = \begin{bmatrix}x\\ x\end{bmatrix}_{\pi}^{\bar{\pi}} \oplus [x \oplus y]^{\pi^{-1}}. \tag{6.9}$$

Braid-parity group:

$$G = \left\langle s \oplus p, \begin{bmatrix}s\\ p\end{bmatrix}_{\pi}^{\bar{\pi}} \right\rangle,$$
$$= \{e, s \oplus p, \begin{bmatrix}s\\ s\end{bmatrix}_{\pi}^{\bar{\pi}} \oplus \begin{bmatrix}p\\ p\end{bmatrix}_{\pi}^{\bar{\pi}},$$
$$\begin{bmatrix}s\\ p\end{bmatrix}_{\pi}^{\bar{\pi}}, \begin{bmatrix}p\\ s\end{bmatrix}_{\pi}^{\bar{\pi}}, \begin{bmatrix}s\\ s\end{bmatrix}_{\pi}^{\bar{\pi}}, \begin{bmatrix}p\\ p\end{bmatrix}_{\pi}^{\bar{\pi}}\}. \tag{6.10}$$

Closed under the ring operations.

* What algebra is this? https://en.wikipedia.org/wiki/Quaternion

Reduction to permutations:

$$\begin{bmatrix}\mathbf{0}\\ x\end{bmatrix}_{\pi}^{\bar{\pi}} = \pi \circ x,$$
$$\begin{bmatrix}x\\ \mathbf{0}\end{bmatrix}_{\pi}^{\bar{\pi}} = \pi \circ \bar{x}. \tag{6.11}$$

VII. TOO MUCH JUNK – SOMEONE TAKE OUT THE TRASH

VIII. DIFFERENTIAL ENCODING

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

$$[\![x, y]\!] \equiv [x, x \oplus y]$$
$$\equiv [x \oplus y, x]. \tag{8.1}$$

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 $\oplus$,

$$[\![x_1, y_1]\!] \oplus [\![x_2, y_2]\!] \Rightarrow [\![x_1 \oplus x_2, y_1 \oplus y_2]\!], \tag{8.2}$$

with the properties,

$$[\![\mathbf{0}, x \oplus y]\!] = [\![x, y]\!],$$
$$[\![x, y]\!] \Rightarrow x = y. \tag{8.3}$$

A. New stuff

Use the convention of differentially encoded private and public information,

$$\Delta_i^{(k)} = [\mathtt{priv}_i^{(k)}, \mathtt{pub}_i]_i, \tag{8.4}$$

where i indexes an encoding and k denotes an individual private party. Given a set of $\{\Delta_i\}_i$,

$$g_{\mathtt{priv}}^{(k)} = \bigcup_i \mathtt{priv}_i^{(k)} \cup \mathtt{pub}_i,$$
$$g_{\mathrm{pub}} = \bigcup_i \mathtt{pub}_i \tag{8.5}$$

are the publicly and privately available objects, acting as group generators for the available algebra of operations under XOR,

$$G_{\mathrm{priv}}^{(k)} \sim \langle g_{\mathtt{priv}}^{(k)} \rangle,$$
$$G_{\mathrm{pub}} \sim \langle g_{\mathtt{pub}} \rangle, \tag{8.6}$$

where the public group is a subgroup of the private group,

$$G_{\mathrm{pub}} \subseteq G_{\mathrm{priv}}^{(k)}. \tag{8.7}$$

1. Differential random codes

Round of entropy exchange

$$A \to B: \quad [s, s \oplus \mathcal{X}_i^A],$$
$$B \to A: \quad [s, s \oplus \mathcal{X}_i^B], \tag{8.8}$$

yields the public information under addition,

$$A, B, \texttt{pub}: \quad [\mathbf{0}, \mathcal{X}_i^A \oplus \mathcal{X}_i^B]. \tag{8.9}$$

Hence the private and public group generators and elements are given by,

$$\begin{aligned} g_{\texttt{priv}}^{A,B} &= \langle s, \mathcal{X}_i^A, \mathcal{X}_i^B \rangle, \\ &= \{\mathbf{0}, s, \mathcal{X}_i^A, \mathcal{X}_i^B, s \oplus \mathcal{X}_i^A, s \oplus \mathcal{X}_i^B, \mathcal{X}_i^A \oplus \mathcal{X}_i^B, s \oplus \mathcal{X}_i^A \oplus \mathcal{X}_i^B\}, \\ &= \mathrm{GF}(2^3), \\ g_{\texttt{pub}} &= \langle s \oplus \mathcal{X}_i^A, s \oplus \mathcal{X}_i^B \rangle \\ &= \{\mathbf{0}, s \oplus \mathcal{X}_i^A, s \oplus \mathcal{X}_i^B, \mathcal{X}_i^A \oplus \mathcal{X}_i^B\}, \\ &= \mathrm{GF}(2^2), \end{aligned} \tag{8.10}$$

which factors into,

$$G_{\texttt{priv}} = G_{\texttt{pub}} \times \langle s \rangle, \tag{8.11}$$

where the private group is the subgroup of the public group invariant under s.

The above implements agreement upon the publicly known shared entropy source where,

$$\tilde{\mathcal{X}}_i = \mathcal{X}_i^A \oplus \mathcal{X}_i^B. \tag{8.12}$$

Differentially encoding two independent shared public entropy sources $\{i, j\}$ over users $\{A, B\}$. Publicly share,

$$\begin{aligned} \Delta_A &= [s \oplus p \oplus \mathcal{X}_i^A, \mathcal{X}_i^A \oplus \mathcal{X}_j^A]_A, \\ \Delta_B &= [s \oplus p \oplus \mathcal{X}_i^B, \mathcal{X}_i^B \oplus \mathcal{X}_j^B]_B, \\ \texttt{pub} &: [s \oplus p \oplus \tilde{\mathcal{X}}_i, \tilde{\mathcal{X}}_i \oplus \tilde{\mathcal{X}}_j], \end{aligned} \tag{8.13}$$

Encode message m,

$$\begin{aligned} \texttt{enc}(m) &= s \oplus m \oplus \tilde{\mathcal{X}}_j, \\ \Delta(m) &= [s \oplus \tilde{\mathcal{X}}_i, m \oplus \tilde{\mathcal{X}}_j], \end{aligned} \tag{8.14}$$

decode $s \oplus m$ given known $\tilde{\mathcal{X}}_i \oplus \tilde{\mathcal{X}}_j]$.

IX. ENTROPY CODES

Entropy codes employ random codewords,

$$\mathcal{X} : x \in \{0, 1\}^n, \tag{9.1}$$

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

$$0 \leq H(\mathcal{X}) \leq 1. \tag{9.2}$$

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

$$\mathbb{P}(x) = \frac{1}{2^n}, \tag{9.3}$$

and are orthogonal,

$$x_i \oplus x_j \neq \mathbf{0}. \tag{9.4}$$

Under differential encoding,

$$[x, x \oplus y], \tag{9.5}$$

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

$$H(x \oplus y) \geq \min(H(x), H(y)). \tag{9.6}$$

Hence parity terms form unique random codewords.

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

A. Collision space

The collision space of an n-bit function,

$$f_n(x) \to y, \quad x, y \in \{0, 1\}^n, \tag{9.7}$$

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

$$\mathrm{Col}(f) = |\{\mathrm{Im}(x) \,|\, \mathrm{Im}(x) = \mathrm{Im}(y \neq x)\}|, \tag{9.8}$$

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

$$\overline{\mathrm{Col}}(f_n) + \mathrm{Col}(f_n) = 2^n, \tag{9.9}$$

paritioning the image space.

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

$$f: x, y \to \begin{bmatrix} x \\ x \end{bmatrix}_\pi^{\bar{\pi}}, \quad x \neq y. \tag{9.10}$$

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,

$$\mathbb{P}(n, k) \approx 1 - \exp\left[-\frac{k^2}{2^{n+1}}\right]. \tag{9.11}$$

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

The collision space of asymmetric objects of the form $\left[\begin{smallmatrix} x \\ x \end{smallmatrix}\right]_\pi^{\bar{\pi}}$ is given by the set of satisfying y such that,

$$\begin{bmatrix} x \\ x \end{bmatrix}_\pi^{\bar{\pi}} = \begin{bmatrix} y \\ y \end{bmatrix}_\pi^{\bar{\pi}}, \tag{9.12}$$

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.

A braided codeword has,

$$c = \begin{bmatrix} x \\ x \end{bmatrix}_{\pi}^{\bar{\pi}}, \tag{9.13}$$

$$\mathrm{Col}(c) = \binom{n}{n/2}, \tag{9.14}$$

unique pre-images as does the function,

$$f(x) = \begin{bmatrix} x \\ x \end{bmatrix}_{\pi}^{\bar{\pi}} \oplus c. \tag{9.15}$$

If a single image is known the entire class of pre-images is trivially obtained under their known automorphisms. The pre-image space, the collision space of $f(x)$, collectively map to a single image, a homomorphism from pre-image space to image space under iterative braiding.

Under re-braiding $f(x)$ maps to a single pre-image in the collision space of $f(f(x))$,

$$\begin{aligned} h(x) &= f(f(x)) \\ &= \begin{bmatrix} f(x) \\ f(x) \end{bmatrix}_{\pi}^{\bar{\pi}} \oplus c \\ &= \begin{bmatrix} x \oplus c \\ x \oplus c \end{bmatrix}_{\pi^2}^{\bar{\pi}^2} \oplus \begin{bmatrix} x \\ x \end{bmatrix}_{\pi}^{\bar{\pi}} \oplus c. \end{aligned} \tag{9.16}$$

Now finding the double pre-image of $h(x)$ requires finding a single pre-image from the collision space of $f(x)$, amongst,

$$|\mathrm{Col}(f(x))| = \binom{n}{n/2}, \tag{9.17}$$

unique pre-images, only one of which has pre-image x. This is in general a many-to-one map where unmapped images are forbidden outcomes with multiplicity corresponding to collision space space.

The space of the available output domain is reduced by its collision space.

Can we make this bijective over the full domains under symmeterisation?

Entropy: group elements as information objects reveal the joint parity (mode 2) of its letters/group generators.

X. PERMUTATION CODES

* Change:

$$R^n_{\mathcal{X}} = (\mathbb{Z}_{2^n}, B_n, \oplus, \star), \tag{10.1}$$

comprising the bitwise binary operators $(\mathbb{Z}_2^n, \oplus)$ and (B_n, π), where B_n is the braid group over n strands, a representation of an ordered pair of permutations, $(\pi, \bar{\pi})$, where $\bar{\pi} \equiv \pi^{-1}$. Defining braids over bit-string elements...

* The algebraic structure is a non-abelian division ring.

* The codeword space for k letters over n bits is,

$$C^k_{\mathcal{X}} = (\mathbb{Z}_2^k, \oplus, \star) \subseteq R^n_{\mathcal{X}}. \tag{10.2}$$

Constraining codewords to balanced bit-strings with Hamming weight,

$$|x|_{\mathrm{wt}} = \sum_{i=1}^{n} x_i = \frac{n}{2}, \quad \forall\, x \in \{1, k\}, \tag{10.3}$$

all codewords are related by permutations,

$$x_i = \pi \circ x_j. \tag{10.4}$$

The permutational relations between codewords are not unique with degeneracies equivalent to the automorphism space given by the action,

$$\begin{aligned} \mathrm{Aut}(x) &= \mathrm{Sym}(\{x_i = 0\}_i) \times \mathrm{Sym}(\{x_i = 1\}_i) \\ \pi &\in S_{n/2} \times S_{n/2}. \end{aligned} \tag{10.5}$$

...

Define a logical group via generators homomorphic to the automorphism space of permutational relations between bitstrings,

$$\begin{aligned} G_{\mathcal{X}} &: \pi \circ x_0 = x_i, \\ G_{\mathrm{L}} &: g_i \cdot x_0 = x_i, \quad g \equiv \mathrm{Aut}(x), \\ \phi &: G_{\mathcal{X}} \to G_{\mathrm{L}}, \end{aligned} \tag{10.6}$$

where ϕ is a surjective homomorphism from permutations onto their automorphisms.

–

The automorphisms of $z = x \oplus y$ are partitioned according z,

$$\mathrm{Aut}(z) = \mathrm{Sym}(\{x_i = y_i\}_i) \times \mathrm{Sym}(\{x_i \neq y_i\}_i). \tag{10.7}$$

The only constraint imposed on z is $|z|_{\mathrm{wt}} = 2\mathbb{Z}$. While the automorphic subspaces are in general not equipartitioned via their symmetry the average case Hamming weight is,

$$\mu(|z|_{\mathrm{wt}}) = \frac{n}{2}. \tag{10.8}$$

The automorphic space of braid pre-images is partitioned over the image parity,

$$c_i = \pi[x]_i \oplus x_i. \tag{10.9}$$

–

We define the algebra,

$$(\mathbb{Z}_2^n \times S_n, \oplus, \pi), \tag{10.10}$$

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

$$\begin{aligned} \pi &\in S_n, \\ \pi \circ x &\equiv x_\pi, \end{aligned} \tag{10.11}$$

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,

$$x_\pi x \equiv x_\pi \oplus x, \tag{10.12}$$

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

$$\begin{aligned} x \oplus (x_\pi \oplus x) &= x_\pi, \\ x_\pi \oplus (x_\pi \oplus x) &= x. \end{aligned} \tag{10.13}$$

Via distributivity we have the additional symmetric property,

$$\pi^{-1} \circ (x_\pi x) = x_{\pi^{-1}} x. \tag{10.14}$$

Complementation:

$$\begin{aligned} \overline{x_\pi \oplus x \oplus c} &= \overline{x_\pi} \oplus x \oplus c \\ &= x_\pi \oplus \overline{x} \oplus c \\ &= x_\pi \oplus x \oplus \overline{c} \\ &= x_\pi \oplus x \oplus c \oplus \mathbb{I}. \end{aligned} \tag{10.15}$$

Hence,

$$\overline{\begin{bmatrix} x \\ x \end{bmatrix}_\pi^{\bar{\pi}}} = \begin{bmatrix} \overline{x} \\ x \end{bmatrix}_\pi^{\bar{\pi}} = \begin{bmatrix} x \\ \overline{x} \end{bmatrix}_\pi^{\bar{\pi}} = \begin{bmatrix} x \\ x \end{bmatrix}_\pi^{\bar{\pi}} \oplus \mathbb{I}. \tag{10.16}$$

Since,

$$\begin{bmatrix} \overline{x} \\ \overline{x} \end{bmatrix}_\pi^{\bar{\pi}} = \begin{bmatrix} x \\ x \end{bmatrix}_\pi^{\bar{\pi}}, \tag{10.17}$$

braid pre-images occur in complimentary pairs, $\{x, \overline{x}\}$.

Since,

$$\begin{aligned} \mathrm{wt}(x_\pi) &= \mathrm{wt}(x), \\ \mathrm{par}(x_\pi) &= \mathrm{par}(x), \end{aligned} \tag{10.18}$$

the parity of a braid is,

$$\begin{aligned} \mathrm{par}\left(\begin{bmatrix} x \\ y \end{bmatrix}_\pi^{\bar{\pi}} \right) &= \mathrm{par}(x) \oplus \mathrm{par}(y), \\ \mathrm{par}\left(\begin{bmatrix} x \\ x \end{bmatrix}_\pi^{\bar{\pi}} \right) &\in 2\mathbb{Z}, \end{aligned} \tag{10.19}$$

where,

$$\begin{aligned} \mathrm{wt}(x) &= \sum_i x_i, \\ \mathrm{par}(x) &= \bigoplus_i x_i, \end{aligned} \tag{10.20}$$

hence while the pre-image space is 2^n the image space is 2^{n-1} owing to the parity constraint and complimentary pairwise pre-images.

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

$$\begin{aligned} \pi \circ [\![x, y]\!] &\equiv [\![x_\pi, y_\pi]\!], \\ [\![x_\pi, y_\pi]\!] &\sim [\![x, y]\!], \\ [\![x, y_\pi]\!] &\not\sim [\![x, y]\!] \\ [\![x_\pi, y]\!] &\not\sim [\![x, y]\!] \end{aligned} \tag{10.21}$$

A. TCF

From the identities we construct a TCF model as,

$$\begin{aligned} f_{\mathrm{tcf}}(x) &= \begin{bmatrix} x \oplus c \\ \overline{x} \oplus \overline{c} \end{bmatrix}_\pi^{\bar{\pi}} = \begin{bmatrix} x \oplus c \\ \overline{x} \oplus c \end{bmatrix}_\pi^{\bar{\pi}} \\ &= \begin{bmatrix} x \\ \overline{x} \end{bmatrix}_\pi^{\bar{\pi}} \oplus \begin{bmatrix} c \\ \overline{c} \end{bmatrix}_\pi^{\bar{\pi}} \\ &= \begin{bmatrix} \overline{x} \\ x \end{bmatrix}_\pi^{\bar{\pi}} \oplus \begin{bmatrix} c \\ \overline{c} \end{bmatrix}_\pi^{\bar{\pi}} \\ &= \begin{bmatrix} x \oplus c \\ \overline{x} \end{bmatrix}_\pi^{\bar{\pi}} \end{aligned} \tag{10.22}$$

which is invariant under $x \leftrightarrow \overline{x}$, defining the paired pre-images.

$$\begin{aligned} f_{\mathrm{tcf}}(x \oplus c) &= \begin{bmatrix} x \\ \overline{x} \end{bmatrix}_\pi^{\bar{\pi}}, \\ f_{\mathrm{tcf}}(x) &= \begin{bmatrix} x \oplus c \\ \overline{x \oplus c} \end{bmatrix}_\pi^{\bar{\pi}} = \begin{bmatrix} \overline{x \oplus c} \\ x \end{bmatrix}_\pi^{\bar{\pi}} = w. \end{aligned} \tag{10.23}$$

implies $\{x \oplus c, \overline{x \oplus c}\}$ form a pre-image pair,

$$f_{\mathrm{tcf}}(x \oplus c) = \tag{10.24}$$

Using the asymmetric hash model,

$$h : x \oplus \begin{bmatrix} y \\ y \end{bmatrix}_\pi^{\bar{\pi}} \xrightarrow{\pi} \begin{bmatrix} x \\ x \end{bmatrix}_\pi^{\bar{\pi}} \oplus y, \tag{10.25}$$

we modify to, Using the asymmetric hash model,

$$f : x \oplus \begin{bmatrix} y \\ \bar{y} \end{bmatrix}_\pi^{\bar{\pi}} \xrightarrow{\pi} \begin{bmatrix} x \\ \bar{x} \end{bmatrix}_\pi^{\bar{\pi}} \oplus y. \tag{10.26}$$

$$f_{\mathrm{tcf}}(x) = \begin{bmatrix} x \oplus c \\ \overline{x} \oplus \overline{c} \end{bmatrix}_\pi^{\bar{\pi}} \tag{10.27}$$

B. Braid codes

Defining the braid operator,

$$\begin{bmatrix} x \\ y \end{bmatrix}_{\pi}^{\bar{\pi}} \equiv x_{\pi^{-1}} \oplus y_{\pi}, \tag{10.28}$$

the braid objects exhibit trapdoor locking effects. Knowing π does not reveal x from the braid $\langle x \rangle_{\pi}^{\pi^{-1}}$. Given $\langle x, y \rangle_{\pi}^{\pi^{-1}}$ and π, knowing either input x or y reveals the other input. Hence, for known π both inputs x and y act as trapdoors.

Braids preserve distributivity over $\oplus$,

$$\begin{aligned} \begin{bmatrix} x \\ x' \end{bmatrix}_{\pi}^{\bar{\pi}} \oplus \begin{bmatrix} y \\ y' \end{bmatrix}_{\pi}^{\bar{\pi}} &= \begin{bmatrix} x \oplus y \\ x' \oplus y' \end{bmatrix}_{\pi}^{\bar{\pi}}, \\ \begin{bmatrix} x \\ x \end{bmatrix}_{\pi}^{\bar{\pi}} \oplus \begin{bmatrix} y \\ y \end{bmatrix}_{\pi}^{\bar{\pi}} &= \begin{bmatrix} x \oplus y \\ x \oplus y \end{bmatrix}_{\pi}^{\bar{\pi}}. \end{aligned} \tag{10.29}$$

A braided differential hash operator takes the form,

$$\Delta(x) = \begin{bmatrix} x \\ h(x) \end{bmatrix}_{\pi}^{\bar{\pi}} = x_{\pi^{-1}} \oplus h_{\pi}(x). \tag{10.30}$$

Preparing,

$$\Delta(s \oplus m_{\pi^{-2}}) = \begin{bmatrix} s \oplus m \\ h(s \oplus m) \end{bmatrix}_{\pi}^{\bar{\pi}}, \tag{10.31}$$

if s is a shared secret and π and $\Delta(s \oplus m)$ are public, sharing $h(s \oplus m)$ allows $\Delta(s \oplus m)$ to be unlocked,

$$\begin{aligned} &\Delta(s \oplus m) \oplus [\pi^{-1} \circ h(s \oplus m)] \\ &\to \langle s, m \rangle_{\pi}^{\pi^{-1}} \to m, \end{aligned} \tag{10.32}$$

where s is known.

XI. CODEWORDS

Codewords are the space of balanced bit-strings, defining the code-space,

$$C_n : x \in \{0,1\}^n,\ \mathrm{wt}(x) = \frac{n}{2},\ n \in 2\mathbb{Z}^+. \tag{11.1}$$

The order of the code-space is,

$$|C_n| = \binom{n}{n/2}. \tag{11.2}$$

All codewords are permutations of one another,

$$x_i = \pi \circ x_j, \quad \pi \in S_n. \tag{11.3}$$

Code-words can be related via their permutational difference or their symmetric (XOR) difference,

$$\begin{aligned} x_i &= \pi \circ x_j, \\ x_i &= c \oplus x_j. \end{aligned} \tag{11.4}$$

Permutations preserve Hamming weight,

$$\mathrm{wt}(\pi \circ x) = \mathrm{wt}(x), \tag{11.5}$$

while bit-strings symmetric differences in general do not. Code-space is therefore automorphic under the action of the symmetric group,

$$\mathrm{Aut}(C_n) = S_n. \tag{11.6}$$

A random permutation $\pi \in \mathcal{X}(S_n)$ applied to a codeword uniformly randomly samples another codeword,

$$\mathcal{X}: \quad \mathrm{Unif}(\pi \in S_n) \circ x_0 \equiv \mathrm{Unif}(x \in C_n). \tag{11.7}$$

Difference: Both sample uniformly over code-space. The former does so by uniformly sampling over transformations of an initial bit-string, while the latter samples directly. Operator sampling inherits the operator's algebraic properties, transforming code-space via a frame of reference change, affording composability and distributivity. For any chosen π the former acts as a pseudo-random function over pre-image space x while the latter is defined as a (non-pseudo) random function.

XII. 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.

$$[s_{\pi} s] \cdot [m_{\pi} h(m)] = [s \cdot m] \cdot [s_{\pi} \cdot h(m_{\pi})] \tag{12.1}$$

We use the notation,

$$\begin{aligned} x_{\pi_1} y_{\pi_2} &\equiv \langle x, y \rangle_{\pi_1}^{\pi_2}, \\ x_{\pi} x_{\pi^{-1}} &\equiv \begin{bmatrix} x \\ \pi \end{bmatrix}_{\pi}^{\bar{\pi}} \pi^{-1}. \end{aligned} \tag{12.2}$$

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 $x \hat{\cdot} y \equiv x \cdot y \equiv xy$ as shorthand.

A. One-way functions

Definition: a function $f : \{0,1\}^* \to \{0,1\}^*$ is one-way if for BPP algorithm F the average case success probability of finding a pseudo-inverse is negligible,

$$\mathbb{P}[f(F(f(x))) = f(x)] < \mathtt{negl}(n), \tag{12.3}$$

for $x \in \mathrm{Unif}(\{0,1\}^n)$.

For random π, $y = x_{\pi}$ is a uniformly distributed sample over the same space as x, randomised under the action of π, independent of x.

Inverting $y = \langle x \rangle_{\pi}^{\pi^{-1}}$: define $x = \hat{\pi} x_0$, where x_0 is a sorted canonical balanced bit-string used as a reference for which all x may be defined relative to permutations,

$$\begin{aligned} x_0 &= \{0^{\times(n/2)}, 1^{\times(n/2)}\}, \\ x &= \hat{\pi} \circ x_0, \end{aligned} \tag{12.4}$$

for some $\hat{\pi}$ which are in general non-unique. The general braid $\langle x \rangle_{\pi}^{\pi^{-1}}$ can be re-expressed in terms of the canonical x_0 as,

$$\begin{aligned} x_\pi x_{\pi^{-1}} &= y, \\ x \cdot x_{\pi^2} &= \pi[y], \quad (\pi\circ) \\ \hat{\pi}(x_0) \cdot [\pi \cdot \hat{\pi}(x_0)] &= y, \quad (x \to \hat{\pi} \cdot x_0) \\ x_0 \cdot [\hat{\pi}^{-1}\pi \cdot \hat{\pi}(x_0)] &= \hat{\pi}^{-1}(y), \quad (\hat{\pi}^{-1}\circ) \\ [\hat{\pi}^{-1} \cdot \pi \cdot \hat{\pi}](x_0) &= x_0 \cdot \hat{\pi}^{-1}(y), \end{aligned} \tag{12.5}$$

* Conjugacy Problem for Permutations (NP-complete): https://chatgpt.com/share/5716548b-5407-4b77-947c-cc84b47c47fc.

* Definition of CPP: Given $\sigma, \tau \in S_n$, find π such that,

$$\pi\sigma\pi^{-1} = \tau. \tag{12.6}$$

* Defining $\tau(x_0) = x_0 \cdot \hat{\pi}^{-1}(y)$ transforms the equation into a common reference frame relative to x_0.

* In the collision-free space only a single solution for x exists and finding π is equivalent to finding x.

* Conjugate space corresponds to collision space, separating into conjugacy classes of different orders of collisions. In the collision free regime there is exactly one conjugate solution.

B. Pre-image resistance

Consider general case of arbitrary Hamming weight. Then,

$$\begin{aligned} \mathrm{Aut}(x) &= \mathrm{Sym}(x^{(0)}) \times \mathrm{Sym}(x^{(1)}) \\ &= S_{n-\mathrm{wt}(x)} \times S_{\mathrm{wt}(x)}. \end{aligned} \tag{12.7}$$

Extreme cases of $\mathrm{wt}(x) = \{0, n\}$ we have,

$$\mathrm{Aut}(x) = \mathrm{Sym}(x), \tag{12.8}$$

and the pre-image space is permutationaly invariant.

Next consider balances bit-strings,

$$\mathcal{X} : x \in \{0,1\}^n, \quad \text{s.t. } w(x) = n/2, \tag{12.9}$$

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,

$$\begin{aligned} \mathrm{Aut}(x) &= \mathrm{Sym}(x^{(0)}) \times \mathrm{Sym}(x^{(1)}) \\ &= S_{n/2} \times S_{n/2}. \end{aligned} \tag{12.10}$$

where,

$$x^{(k)} = \{i \,|\, x_i = k\}_{i \in x}. \tag{12.11}$$

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

$$\frac{|\mathrm{Sym}(x)|}{|\mathrm{Aut}(x)|} = \frac{n!}{(\frac{n}{2})!(\frac{n}{2})!} = \binom{n}{n/2}. \tag{12.12}$$

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 $w(x_\pi) = w(x) = n/2$.

In general, the pre-image space has collision space multiplicity (i.e automorphism space),

$$|\mathrm{Aut}(x)| = \binom{n}{\mathrm{wt}(x)}. \tag{12.13}$$

Let,

$$x_\pi x = c, \tag{12.14}$$

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

Consider the table,

$$[x_\pi, \tilde{x}, \tilde{x} \oplus \tilde{c}]_i, \tag{12.15}$$

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. (**??**) acts distributively over $\oplus$,

$$x \oplus y \xrightarrow{\pi} \begin{bmatrix} x \\ x \end{bmatrix}_{\pi}^{\bar{\pi}} \oplus \begin{bmatrix} y \\ y \end{bmatrix}_{\pi}^{\bar{\pi}}. \tag{12.16}$$

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,

$$h : x \oplus \begin{bmatrix} y \\ y \end{bmatrix}_{\pi}^{\bar{\pi}} \xrightarrow{\pi} \begin{bmatrix} x \\ x \end{bmatrix}_{\pi}^{\bar{\pi}} \oplus y, \tag{12.17}$$

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.

–

Using the model,

$$h(x) = \begin{bmatrix} x \\ x \end{bmatrix}_{\pi}^{\bar{\pi}} \oplus c, \tag{12.18}$$

we have,

$$h(x \oplus y) = h(x) \oplus h(y) \oplus c, \tag{12.19}$$

and,

$$[h(x \oplus y), h(x) \oplus h(y)] = c, \tag{12.20}$$

is the additive commutator (or distributor) of the action of h over its pre-image space.

E. TCF model

Using the asymmetric model,

$$f : x \oplus \begin{bmatrix} x_0 x_1 \\ x_0 x_1 \end{bmatrix}_{\pi}^{\bar{\pi}} \xrightarrow{\pi} \begin{bmatrix} x \\ x \end{bmatrix}_{\pi}^{\bar{\pi}} \oplus \begin{bmatrix} x_0 \\ x_1 \end{bmatrix}_{\pi}^{\bar{\pi}}. \tag{12.21}$$

$$\begin{aligned} f(x_0) &= x_0^{\pi} x_0^{\pi^{-1}} \cdot x_0^{\pi} x_1^{\pi^{-1}} = x_0^{\pi^{-1}} x_1^{\pi^{-1}}, \\ f(x_1) &= x_1^{\pi} x_1^{\pi^{-1}} \cdot x_0^{\pi} x_1^{\pi^{-1}} = x_0^{\pi} x_1^{\pi}. \end{aligned} \tag{12.22}$$

$$\begin{aligned} f(x) &= x^{\pi} x \cdot x_0 x_1, \\ f(x_0) &= x_0^{\pi} x_0^{\pi^{-1}} \cdot x_0^{\pi} x_1^{\pi^{-1}} = x_0^{\pi^{-1}} x_1^{\pi^{-1}}, \\ f(x_1) &= x_1^{\pi} x_1^{\pi^{-1}} \cdot x_0^{\pi} x_1^{\pi^{-1}} = x_0^{\pi} x_1^{\pi}. \end{aligned} \tag{12.23}$$

XIII. ENCRYTPION

Given secrets $\{x, m\}$ with respective public objects $\{\begin{bmatrix} x \\ x \end{bmatrix}_{\pi}^{\bar{\pi}}, \begin{bmatrix} m \\ m \end{bmatrix}_{\pi}^{\bar{\pi}}\}$, public information can be expressed,

$$\begin{aligned} \begin{bmatrix} x \\ x \end{bmatrix}_{\pi}^{\bar{\pi}} \cdot \begin{bmatrix} m \\ m \end{bmatrix}_{\pi}^{\bar{\pi}} &= \begin{bmatrix} m \cdot x \\ m \cdot x \end{bmatrix}_{\pi}^{\bar{\pi}} \\ &= x_{\pi} x_{\pi^{-1}} \cdot m_{\pi} m_{\pi^{-1}}. \end{aligned} \tag{13.1}$$

Now the solution revealing m_{π} is,

$$\texttt{sol} = x_{\pi} x_{\pi^{-1}} m_{\pi^{-1}}, \tag{13.2}$$

hence unlocking m under known π.

Applying known π yields public,

$$x_{\pi^2} x \cdot m_{\pi^2} m. \tag{13.3}$$

Provision of,

$$\texttt{sig} = m_{\pi^2} \cdot x_{\pi^2} \cdot x, \tag{13.4}$$

affords,

$$\texttt{sig}^{\oplus \pi} \circ \langle x \oplus m \rangle_{\pi^{-1}}^{\pi} = m. \tag{13.5}$$

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

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

$$[x, x \oplus y] \xrightarrow{\pi} [x, x \oplus y_{\pi}]. \tag{13.6}$$

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,

$$[\![x, y]\!] \Rightarrow [\![x_{\pi} x, y_{\pi} y]\!]. \tag{13.7}$$

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

$$\begin{aligned} [\![s, p]\!] &= [s, s \oplus p], \\ [s, s \oplus p] &= [p \oplus s, p], \\ [\![s_{\pi} s, p_{\pi} p]\!] &= [s \oplus p, s_{\pi} \oplus p_{\pi}], \end{aligned} \tag{13.8}$$

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

$$[\![s, m]\!] = [s, s \oplus m]. \tag{13.9}$$

Under composition this implies,

$$[s, s \oplus p] \oplus [s, s \oplus m] = [\mathbf{0}, p \oplus m]. \tag{13.10}$$

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,

$$\begin{aligned} &s_\pi s, \\ &p_\pi p. \end{aligned} \tag{13.11}$$

Using the permuted differential codes we have the identity,

$$[s, s{\oplus}p]{\oplus}[s, s_\pi{\oplus}m_\pi] = [\mathbf{0}, s_\pi s{\oplus}m_\pi p]. \tag{13.12}$$

$$[s, s{\oplus}p]{\oplus}[p, p_\pi{\oplus}m_\pi] = [\mathtt{pk}, p_\pi s{\oplus}m_\pi p]. \tag{13.13}$$

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

XIV. OLD

We define asymmetric key-pairs via their differential encoding,

$$\mathtt{sk}, \mathtt{pk} \in \{0,1\}^n, \tag{14.1}$$

$$\begin{aligned} \mathtt{kp} &= [\![\mathtt{sk}, \mathtt{pk}]\!] \\ &= [\mathtt{sk}, \mathtt{sk}{\oplus}\mathtt{pk}], \end{aligned} \tag{14.2}$$

XV. DIGITAL SIGNATURES

Generate signature,

$$\begin{aligned} \mathtt{sig}(m, \mathtt{sk}) &= [\![\mathtt{sk}, \mathtt{pk}]\!]{\oplus}[\![\mathtt{sk}, m]\!] \\ &= [\![\mathbf{0}, \mathtt{pk}{\oplus}m]\!] \\ &= \mathtt{pk}{\oplus}m. \end{aligned} \tag{15.1}$$

Verify signature,

$$\begin{aligned} \mathtt{ver}(\mathtt{sig}, \mathtt{pk}) &= \Delta(m, \mathtt{pk}){\oplus}\Delta() \\ &= \mathtt{pk}. \end{aligned} \tag{15.2}$$

XVI. ASYMMETRIC ENCRYPTION

Encode,

$$\begin{aligned} \tilde{m} &= \mathtt{enc}(m, \mathtt{pk}) \\ &= [\![m, \mathtt{sk}]\!] \\ &= \mathtt{sk}{\oplus}m. \end{aligned} \tag{16.1}$$

Decode,

$$\begin{aligned} \mathtt{dec}(\tilde{m}, \mathtt{sk}) &= \tilde{m}{\oplus}\bar{\mathtt{kp}} \\ &= \mathtt{enc}(\mathtt{sk}, m){\oplus}[\![\mathtt{pk}, \mathtt{sk}]\!] \\ &= \mathtt{sk}{\oplus}m. \end{aligned} \tag{16.2}$$

XVII. AUTHENTICATION

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

XVIII. KEY REUSE

Consider multiple messages m_i signed by the same public key,

$$\mathtt{enc}(m_i, \mathtt{pk}) = \mathtt{pk}{\oplus}m_i, \tag{18.1}$$

Verify signature,

$$\begin{aligned} \mathtt{ver}(\mathtt{sig}, \mathtt{pk}) &= \Delta(m, \mathtt{pk}){\oplus}\Delta() \\ &= \mathtt{pk}. \end{aligned} \tag{18.2}$$

A. Key-establishment & secret sharing

XIX. OLD STUFF

XX. DIFFERENTIAL HASH CODES

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

$$[x, x{\oplus}y]_{\oplus}, \tag{20.1}$$

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,

$$\begin{aligned} \Delta(x) &= h(x){\oplus}x, \\ \Delta_\pi(x) &= h(x_\pi){\oplus}x, \end{aligned} \tag{20.2}$$

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,

$$\begin{aligned} h(x) &= \Delta(x){\oplus}x, \\ x &= \Delta(x){\oplus}h(x). \end{aligned} \tag{20.3}$$

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,

$$[x, \Delta(x)]_{\oplus} = [x, h(x){\oplus}x]_{\oplus}, \tag{20.4}$$

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,

$$[h(x), \Delta(x)] = [h(x), h(x) \oplus x], \tag{20.5}$$

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,

$$\begin{aligned} \Delta(x \oplus y) &= h(x \oplus y) \oplus x \oplus y \\ \Delta(x) \oplus \Delta(y) &= h(x) \oplus h(y) \oplus x \oplus y. \end{aligned} \tag{20.6}$$

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

$$\Delta(x \oplus y) \oplus \Delta(x) \oplus \Delta(y) = h(x \oplus y) \oplus h(x) \oplus h(y), \tag{20.7}$$

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

Standard differential codes are composable,

$$\begin{aligned} &[x, x \oplus y)]_{\oplus} \oplus [x', x' \oplus y']_{\oplus} \\ &\sim [x \oplus x', x \oplus y \oplus x' \oplus y']_{\oplus}. \end{aligned} \tag{20.8}$$

For differential hash codes,

$$\begin{aligned} &[x, \Delta(x)]_{\oplus}, \\ &[y, \Delta(y)]_{\oplus}, \end{aligned} \tag{20.9}$$

we have distinct composition rules,

$$\begin{aligned} &[x \oplus y, \Delta(x \oplus y)]_H, \\ &[x \oplus y, \Delta(x) \oplus \Delta(y))]_{\oplus}. \end{aligned} \tag{20.10}$$

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,

$$\begin{aligned} &\pi(x \oplus y) = \pi(x) \oplus \pi(y), \\ &\pi(x) \oplus y \neq x \oplus \pi(y), \end{aligned} \tag{20.11}$$

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

$$x \oplus y = y \oplus x. \tag{20.12}$$

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

XXI. ASYMMETRIC CODES

We define key-pairs as,

$$\begin{aligned} \mathtt{sk} &\in \{0,1\}^n, \\ \mathtt{pk} &= \{\mathtt{pk}_\Delta, \mathtt{pk}_\pi\}, \\ \mathtt{pk}_\Delta &= \Delta(\mathtt{sk}), \\ \mathtt{pk}_\pi &\in S_n, \end{aligned} \tag{21.1}$$

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.

$$[m_\pi \oplus s_\pi, \Delta_\pi(m_\pi \oplus s_\pi)], \tag{21.2}$$

$$\begin{aligned} \Delta_\pi(m_\pi \oplus s_\pi) &= \Delta_\pi(m_\pi)\Delta_\pi(s_\pi) \\ &\oplus h(m_\pi) \oplus h(s_\pi) \oplus h(m_\pi \oplus s_\pi) \end{aligned} \tag{21.3}$$

XXII. DIGITAL SIGNATURES

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

$$\mathtt{sig}_\pi(m) = h(m_\pi \oplus \mathtt{sk}_\pi). \tag{22.1}$$

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

$$\mathtt{sig}_\pi(m) \oplus \Delta(m) \oplus \Delta(\mathtt{sk}_\pi) = h(m). \tag{22.2}$$

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

$$\mathtt{sk}_\pi \equiv \mathtt{sk} \oplus h(\mathtt{pk}), \tag{22.3}$$

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

$$\begin{aligned} h(\mathtt{sk}) \oplus \Delta(\mathtt{sk}) &= \mathtt{sk}, \\ h(\mathtt{sk}_\pi) \oplus \Delta(\mathtt{sk}) &\neq \mathtt{sk}. \end{aligned} \tag{22.4}$$

XXIII. ENCRYPTION

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

$$\mathtt{enc}(m) = \{\Delta(m), h(m)\}. \tag{23.1}$$

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

$$\Delta(s_\pi) \oplus \Delta(m) \oplus h(m) = m. \tag{23.2}$$

A. Multi-sigs

Signatures are commutative, additive and composable.

$$\texttt{sig}_\pi(m) = \Delta(s_\pi) \oplus h(m), \tag{23.3}$$

B. Key-establishment & secret sharing

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

$$\texttt{key} = h(\texttt{sk} \oplus m), \tag{23.4}$$

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.

XXIV. NOTES

* The hash collision space translates to decoding failure.

Differentially encoded tuples are composable under bitwise XOR,

$$[x, \Delta(x)] \oplus [y, \Delta(y)] = [x \oplus y, \Delta(x) \oplus \Delta(y)], \tag{24.1}$$

via the commutativity of $\oplus$.

Compare above rhs,

$$h(x \oplus y) \oplus x \oplus y = h(x \oplus y) \oplus h(x) \oplus h(y). \tag{24.2}$$

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, $\oplus$,

$$\pi(x) \oplus \pi(y) = \pi(x \oplus y). \tag{24.3}$$

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

$$[x, x \oplus y] \sim [\pi(x), \pi(x \oplus y)], \tag{24.4}$$

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

$$[x, x \oplus y] \not\sim [\pi(x), x \oplus y]. \tag{24.5}$$

We'll employ the shorthand,

$$\pi \circ [x, y] = [\pi(x), \pi(y)]., \tag{24.6}$$

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

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

$$\pi(h(x)) \neq h(\pi(x)). \tag{24.7}$$

$$\Delta(\pi(x \oplus y)) = h(\pi(x \oplus y)) \oplus \pi(x) \oplus \pi(y). \tag{24.8}$$

A. Symmetric encryption

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

$$\begin{aligned} s &\in \{0,1\}^n, , \\ \mathcal{X}_i &\in \{0,1\}^n, \\ &[s, s \oplus \mathcal{X}_i]. \end{aligned} \tag{24.9}$$

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

$$\begin{aligned} \texttt{Priv} : &\{s, \mathcal{X}_i, \mathcal{X}_j, \mathcal{X}_{i,j}\} \sim \langle s, \mathcal{X}_i, \mathcal{X}_j \rangle \\ &\sim \mathrm{GF}(2^3), \\ \texttt{Pub} : &\{s \oplus \mathcal{X}_i, s \oplus \mathcal{X}_j, \mathcal{X}_{i,j}\} \\ &\sim \langle s \oplus \mathcal{X}_i, s \oplus \mathcal{X}_j \rangle \sim \langle \mathcal{X}_i, \mathcal{X}_j \rangle \times \langle s \rangle \\ &\sim \mathrm{GF}(2^2), \end{aligned} \tag{24.10}$$

where,

$$\mathcal{X}_{i,j} = \mathcal{X}_i \oplus \mathcal{X}_j, \tag{24.11}$$

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,

$$[s, s \oplus m \oplus \mathcal{X}_i], \tag{24.12}$$

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

Publicly communicating the differential term, the spaces of private and public information form groups with group generators given by privately held and publicly shared objects,

$$\begin{aligned} G_{\text{priv}} &\sim \mathrm{GF}(2^4) \sim \langle s, m, \mathcal{X}_i, \mathcal{X}_j \rangle, \\ G_{\text{pub}} &\sim \mathrm{GF}(2^3) \sim \langle s \oplus \mathcal{X}_i, s \oplus \mathcal{X}_j, s \oplus m \rangle, \end{aligned} \tag{24.13}$$

with respective group elements under binary addition (XOR),

$$\begin{aligned} G_{\text{pub}} : &\{\emptyset, \mathcal{X}_{i,j}, \\ &s \oplus \mathcal{X}_i, s \oplus \mathcal{X}_j, \\ &m \oplus \mathcal{X}_i, m \oplus \mathcal{X}_j, \\ &s \oplus m \oplus \mathcal{X}_{i,j}\}, \\ G_{\text{priv}} : &\{s, m, \mathcal{X}_i, \mathcal{X}_j\} \times \langle G_{\text{pub}} \rangle, \end{aligned} \tag{24.14}$$

Now public information reveals $m \oplus s$ but not m or s individually whereas the additional private information does.

Define a pair of secrets $\{s, p\}$,

$$[s, s \oplus p], \tag{24.15}$$

let us instead encode m against p,

$$[p, m \oplus p \oplus \mathcal{X}_i]. \tag{24.16}$$

Then the public and private groups are generated by,

$$\begin{aligned} G_{\text{priv}} &\sim \langle s, p, m, \mathcal{X}_i, \mathcal{X}_j \rangle, \\ G_{\text{pub}} &\sim \langle s \oplus \mathcal{X}_i, s \oplus \mathcal{X}_j, p \oplus m \oplus \mathcal{X}_i \rangle \\ &= \{\mathcal{X}_{i,j}, s \oplus \mathcal{X}_i, s \oplus \mathcal{X}_j, \\ &\quad p \oplus m \oplus \mathcal{X}_i, p \oplus m \oplus \mathcal{X}_j, \\ &\quad s \oplus p \oplus m, s \oplus p \oplus m \oplus \mathcal{X}_{i,j}\}. \end{aligned} \tag{24.17}$$

–

HERE:

Sending the encoded message $m \oplus s \oplus \mathcal{X}_j$ (one-time-pad encoding against $s \oplus \mathcal{X}_{i,j}$) we obtain the groups,

$$\begin{aligned} \texttt{Priv} &: \langle s, p, m, \mathcal{X}_i, \mathcal{X}_j \rangle, \\ \texttt{Pub} &: \langle s \oplus \mathcal{X}_i, p \oplus \mathcal{X}_j, m \oplus s \oplus \mathcal{X}_{i,j} \rangle \\ &\sim \{s \oplus \mathcal{X}_i, p \oplus \mathcal{X}_j, s \oplus p \oplus \mathcal{X}_{i,j}, \\ &\quad m \oplus \mathcal{X}_j, m \oplus p \oplus \mathcal{X}_i, \\ &\quad \} \end{aligned} \tag{24.18}$$

–

$$\begin{aligned} A &\to [s_i, s_i \oplus \mathcal{X}_i], \\ B &\to [p_j, p_j \oplus \mathcal{X}_j], \\ A, B &\leftarrow [s_i \oplus p_j, s_i \oplus p_j \oplus \mathcal{X}_{i,j}], \\ E &\leftarrow \{s_i \oplus \mathcal{X}_i, p_j \oplus \mathcal{X}_j, s_i \oplus p_j \oplus \mathcal{X}_{i,j}\}, \\ A &\to [p_j, p_j \oplus m \oplus \mathcal{X}_i], \\ B &\leftarrow m, \\ E &\leftarrow \{s \oplus \mathcal{X}_i, s \oplus \mathcal{X}_{i,j}, \mathcal{X}_j, \\ &\quad p \oplus m \oplus \mathcal{X}_i, p \oplus s \oplus m\} \end{aligned} \tag{24.19}$$

–

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,

$$s \oplus x_i \oplus s \oplus x_j = x_i \oplus x_j, \tag{24.20}$$

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,

$$I(X;Y) = H(X) + H(Y) - H(X,Y). \tag{24.21}$$

Let,

$$\begin{aligned} X &\equiv \{s, x\}, \\ Y &\equiv s \oplus x, \end{aligned} \tag{24.22}$$

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

$$\begin{aligned} H(X) &\leq H(Y) \leq 1, \\ H(X,Y) &= 2H(X), \\ I(X;Y) &= 2H(X) - H(X,Y) \\ &= 0, \\ I(E) &= I(X_s;Y) = I(X_x;Y) \\ &= H(X) + H(Y) - H(X,Y) \\ &= H(Y) - H(X) \\ &\leq 1 - H(X). \end{aligned} \tag{24.23}$$

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,

$$\begin{aligned} b_{A,B} &\in \{Z \equiv 0, X \equiv 1\}, \\ m_{A,B} &\in \{0, 1\}. \end{aligned} \tag{24.24}$$

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

$$m_A \oplus m_B = p_b, \quad (b_A = b_B,\, p_b \in \{0, 1\}). \tag{24.25}$$

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.

XXV. ASYMMETRIC CODES

We define key-pairs as,

$$\begin{aligned} \texttt{sk} &\in \{0, 1\}^n, \\ \texttt{pk} &= \{\texttt{pk}_\Delta, \texttt{pk}_\pi\}, \\ \texttt{pk}_\Delta &= \Delta(\texttt{sk}), \\ \texttt{pk}_\pi &\in S_n, \end{aligned} \tag{25.1}$$

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.

$$[m_\pi \oplus s_\pi, \Delta_\pi(m_\pi \oplus s_\pi)], \tag{25.2}$$

$$\begin{aligned}\Delta_\pi(m_\pi \oplus s_\pi) = \Delta_\pi(m_\pi)\Delta_\pi(s_\pi) \\ \oplus h(m_\pi) \oplus h(s_\pi) \oplus h(m_\pi \oplus s_\pi)\end{aligned} \tag{25.3}$$

XXVI. DIGITAL SIGNATURES

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

$$\mathtt{sig}_\pi(m) = h(m_\pi \oplus \mathtt{sk}_\pi). \tag{26.1}$$

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

$$\mathtt{sig}_\pi(m) \oplus \Delta(m) \oplus \Delta(\mathtt{sk}_\pi) = h(m). \tag{26.2}$$

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

$$\mathtt{sk}_\pi \equiv \mathtt{sk} \oplus h(\mathtt{pk}), \tag{26.3}$$

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

$$\begin{aligned} h(\mathtt{sk}) \oplus \Delta(\mathtt{sk}) &= \mathtt{sk}, \\ h(\mathtt{sk}_\pi) \oplus \Delta(\mathtt{sk}) &\neq \mathtt{sk}. \end{aligned} \tag{26.4}$$

XXVII. ENCRYPTION

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

$$\mathtt{enc}(m) = \{\Delta(m), h(m)\}. \tag{27.1}$$

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

$$\Delta(s_\pi) \oplus \Delta(m) \oplus h(m) = m. \tag{27.2}$$

A. Multi-sigs

Signatures are commutative, additive and composable.

$$\mathtt{sig}_\pi(m) = \Delta(s_\pi) \oplus h(m), \tag{27.3}$$

B. Key-establishment & secret sharing

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

$$\mathtt{key} = h(\mathtt{sk} \oplus m), \tag{27.4}$$

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.

XXVIII. NOTES

* The hash collision space translates to decoding failure.

Differentially encoded tuples are composable under bit-wise XOR,

$$[x, \Delta(x)] \oplus [y, \Delta(y)] = [x \oplus y, \Delta(x) \oplus \Delta(y)], \tag{28.1}$$

via the commutativity of $\oplus$.

Compare above rhs,

$$h(x \oplus y) \oplus x \oplus y = h(x \oplus y) \oplus h(x) \oplus h(y). \tag{28.2}$$

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, $\oplus$,

$$\pi(x) \oplus \pi(y) = \pi(x \oplus y). \tag{28.3}$$

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

$$[x, x \oplus y] \sim [\pi(x), \pi(x \oplus y)], \tag{28.4}$$

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

$$[x, x \oplus y] \not\sim [\pi(x), x \oplus y]. \tag{28.5}$$

We'll employ the shorthand,

$$\pi \circ [x, y] = [\pi(x), \pi(y)]., \tag{28.6}$$

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

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

$$\pi(h(x)) \neq h(\pi(x)). \tag{28.7}$$

$$\Delta(\pi(x \oplus y)) = h(\pi(x \oplus y)) \oplus \pi(x) \oplus \pi(y). \tag{28.8}$$

XXIX. 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,

$$\begin{aligned}&[h(m), m \oplus h(m)]_{\oplus} = [h(m), \Delta(m)]_{\oplus},\\ &[s \oplus h(m), \Delta(s) \oplus \Delta(m)]_{\oplus},\\ &[s \oplus h(m), \Delta(s \oplus h(m))]_{\oplus}\\ &[s \oplus h(m), \Delta(s) \oplus \Delta(m) \oplus h(s \oplus h(m)) \oplus h(m) \oplus h(s)]_{\oplus},\end{aligned} \tag{29.1}$$

if $\Delta(m)$ is public.

This does not unlock from known π,

$$[h(\pi(m)), \pi(m) \oplus h(m)]_{\oplus} \tag{29.2}$$

This unlocks,

$$[h(\pi(s \oplus m)), \Delta(s \oplus m)]_{\oplus} \tag{29.3}$$

Want to decrypt with,

$$\begin{aligned}h(\pi(s \oplus h(m)) = \Delta(\pi(s \oplus h(m)))\\ \oplus \pi(s) \oplus \pi(h(m)).\end{aligned} \tag{29.4}$$

$$[h(m), \Delta(\pi(m)) \oplus \Delta(\pi(s))]_{\oplus} \tag{29.5}$$

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

$$\Delta(\pi(m)) \oplus \Delta(\pi(s)) = \Delta \tag{29.6}$$

$$\begin{aligned}&\mathtt{sig} = \Delta(m) = \pi(m) \oplus h(\pi(m))\\ &\pi(m) = \mathtt{sig} \oplus h(\pi(m))\end{aligned} \tag{29.7}$$

A. Digital signatures

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

$$\mathtt{sig}(m) = \{\pi(m), \Delta(m), h(\mathtt{sk} \oplus m)\}. \tag{29.8}$$

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

$$\pi(\mathtt{sk} \oplus m) = \pi(\Delta(\mathtt{sk}) \oplus \Delta(m) \oplus h(m) \oplus (h(\mathtt{sk})) \tag{29.9}$$

The encodings,

$$\begin{aligned}[\mathtt{sk}, \Delta(\mathtt{sk})]_H, [h(\mathtt{sk}), \Delta(\mathtt{sk})]_{\oplus}\\ [m, \Delta(m)]_H, [h(m), \Delta(m)]_{\oplus}\\ [m \oplus \mathtt{sk} \oplus h(m) \oplus h(\mathtt{sk}), \Delta(m) \oplus \Delta(\mathtt{sk})]_{\oplus}\end{aligned} \tag{29.10}$$

enable verification by hashing the first term.

The differential code,

$$[\mathtt{sk} \oplus m, \Delta(\mathtt{sk} \oplus m)] \tag{29.11}$$

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

$$[\pi(\mathtt{sk} \oplus m), \Delta(\pi(\mathtt{sk} \oplus m))]_H \tag{29.12}$$

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

$$\begin{aligned}[h(\mathtt{sk} \oplus m), \Delta(\mathtt{sk} \oplus m)]_{\oplus}\\ [h(\pi(\mathtt{sk} \oplus m)), \Delta(\pi(\mathtt{sk} \oplus m))]_{\oplus}\end{aligned} \tag{29.13}$$

are both verifiable from direct multiplication.

B. Blah

Hence,

$$[h(\pi(\mathtt{sk} \oplus m)), \Delta(\pi(\mathtt{sk} \oplus m))] \tag{29.14}$$

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

$$\begin{aligned}t_2 &= \Delta(\pi(\mathtt{sk} \oplus m))\\ &= \Delta(\pi(\mathtt{sk})) \oplus \Delta(\pi(m))\\ &\oplus h(\pi(\mathtt{sk} \oplus m)) \oplus h(\pi(\mathtt{sk})) \oplus h(\pi(m))\end{aligned} \tag{29.15}$$

Let signature be,

$$\mathtt{sig} = h(\pi(\mathtt{sk} \oplus m)), \tag{29.16}$$

$$\mathtt{ver} = \Delta(\pi(\mathtt{sk} \oplus m)) \tag{29.17}$$

$$[\mathtt{sig}, \mathtt{ver}] \tag{29.18}$$

XXX. BLAH

$$\begin{aligned}\Delta(\pi(\mathtt{sk} \oplus m)) &= h(\pi(\mathtt{sk} \oplus m)) \oplus \pi(\mathtt{sk} \oplus m)\\ &= h(\pi(\mathtt{sk}) \oplus \pi(m)) \oplus \pi(\mathtt{sk} \oplus m).\\ \Delta(\pi(\mathtt{sk})) \oplus \Delta(\pi(m)) &= \pi(\mathtt{sk}) \oplus \pi(m)\\ &\oplus h(\pi(\mathtt{sk})) \oplus h(\pi(m)).\\ \Delta(\pi(\mathtt{sk} \oplus m)) &= \Delta(\pi(\mathtt{sk})) \oplus \Delta(\pi(m))\\ &\oplus h(\pi(\mathtt{sk} \oplus m)) \oplus h(\pi(\mathtt{sk})) \oplus h(\pi(m))\end{aligned} \tag{30.1}$$

$$\Delta(x \oplus \pi(x)) = h(x \oplus \pi(x)) \oplus x \oplus \pi(x). \tag{30.2}$$

$$\Delta(\mathtt{sk} \oplus m \oplus \pi(\mathtt{sk} \oplus m)) = h(\mathtt{sk} \oplus m \oplus \pi(\mathtt{sk}) \oplus \pi(m)) \\ \mathtt{sk} \oplus m \oplus \pi(\mathtt{sk}) \oplus \pi(m). \tag{30.3}$$

The differential permuted code,

$$[\pi(\mathtt{sk} \oplus m), \Delta(\pi(\mathtt{sk} \oplus m))] \tag{30.4}$$

is verifiable from,

$$h(\pi(\mathtt{sk} \oplus m)) \tag{30.5}$$

The differential permuted code,

$$[\pi(\mathtt{sk} \oplus m), \Delta(\pi(\mathtt{sk} \oplus m))] \tag{30.6}$$

is verifiable from,

$$h(\pi(\mathtt{sk} \oplus m)) \oplus \pi(\mathtt{sk} \oplus m) = bob \\ \oplus h(\pi(\mathtt{sk} \oplus m)) \oplus h(\pi(\mathtt{sk})) \oplus h(\pi(m)) \\ = \tag{30.7}$$

While,

$$[\mathtt{sk} \oplus m, \Delta(\mathtt{sk} \oplus m)] \\ = [\mathtt{sk} \oplus m, \Delta(\mathtt{sk}) \oplus \Delta(m) \\ \oplus h(\mathtt{sk} \oplus m) \oplus h(\mathtt{sk}) \oplus h(m)], \tag{30.8}$$

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

Bob's test passes if,

$$\pi(\mathtt{sk} \oplus m) = \Delta(\pi(\mathtt{sk})) \oplus \Delta(\pi(m)) \oplus h(\pi(\mathtt{sk})) \oplus h(\pi(m)) \tag{30.9}$$

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

Similarly,

$$[\mathtt{sk} \oplus m, \Delta(\mathtt{sk} \oplus m)] \\ = [\mathtt{sk} \oplus m, h(\mathtt{sk} \oplus m) \oplus h(\mathtt{sk}) \oplus h(m)], \tag{30.10}$$

are verifiable from hashing the first term, while,

$$[\pi(\mathtt{sk} \oplus m), \Delta(\mathtt{sk} \oplus m)] \\ = [\pi(\mathtt{sk} \oplus m), h(\mathtt{sk} \oplus m) \oplus h(\mathtt{sk}) \oplus h(m)], \tag{30.11}$$

are verifiable by hashing the unpermuted first term. Equivalently,

$$[\pi(\mathtt{sk} \oplus m), \Delta(\mathtt{sk} \oplus m)] \\ [\pi(\mathtt{sk} \oplus m), \Delta(\mathtt{sk}) \oplus \Delta(m) \oplus h(\mathtt{sk} \oplus m) \oplus h(\mathtt{sk}) \oplus h(m)] \tag{30.12}$$

is verifiable by hashing the unpermuted first term.

Using,

$$\Delta(\mathtt{sk}) \oplus \Delta(m) = \Delta(\mathtt{sk} \oplus m) \oplus h(\mathtt{sk} \oplus m) \oplus h(\mathtt{sk}) \oplus h(m), \tag{30.13}$$

Ver:

$$\{\pi(m) \oplus \pi(\mathtt{sk}), \Delta(m) \oplus \Delta(\mathtt{sk})\} \\ \{\pi(m \oplus \mathtt{sk}), \Delta(m) \oplus \Delta(\mathtt{sk})\} \tag{30.14}$$

We have the relationships,

$$\Delta(\mathtt{sk}) \oplus \Delta(m) = \mathtt{sk} \oplus m \oplus h(\mathtt{sk}) \oplus h(m), \\ \Delta(\mathtt{sk} \oplus m) = h(\mathtt{sk} \oplus m) \oplus \mathtt{sk} \oplus m, \tag{30.15}$$

Under composition we obtain,

$$\Delta(\mathtt{sk} \oplus m) \oplus \Delta(\mathtt{sk}) \oplus \Delta(m) = h(\mathtt{sk} \oplus m) \oplus h(\mathtt{sk}) \oplus h(m). \tag{30.16}$$

The differential encoding,

$$[h(\mathtt{sk} \oplus m), \Delta(\mathtt{sk} \oplus m)], \tag{30.17}$$

Consider the permuted differential code,

$$[\pi(\mathtt{sk} \oplus m), \pi(\Delta(\mathtt{sk})) \oplus \pi(\Delta(m))]. \tag{30.18}$$

we equivalently obtain,

$$[\Delta(\mathtt{sk} \oplus m) \oplus h(\mathtt{sk} \oplus m), \Delta(\mathtt{sk}) \oplus \Delta(m)]. \tag{30.19}$$

Therefore if Alice provides the signature,

$$\mathtt{sig}(m) = [\pi(\mathtt{sk} \oplus m), h(\mathtt{sk} \oplus m)], \tag{30.20}$$

Bob is able to verify via,

$$[\mathtt{sig}(m), \Delta(\mathtt{sk}) \oplus \Delta(m)]. \tag{30.21}$$

REFERENCES